International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

# Comparative Analysis of YOLOv8 and SSD for Real-Time Object Detection in Driving Environments

Saurabh Mapari * and R. S. Paswan

*Department of Computer Engineering SCTR's Pune Institute of Computer Technology Pune, India.*

International Journal of Science and Research Archive, 2026, 19(02), 135-140

Publication history: Received on 23 March 2026; revised on 02 May 2026; accepted on 05 May 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.2.0919

## Abstract

Real-time object detection is crucial factor for various intelligent transportation systems, surveillance systems, and the self-driving car technologies. This research aims to evaluate the performance of two most commonly used object detection algorithms, YOLOv8 and Single Shot Multi Box Detector (SSD), with the help of the BDD100K dataset. Both algorithms are trained and evaluated under the same experimental settings to compare their performance effectively. Various performance evaluation metrics were used to compare the performance of the YOLOv8 and SSD algorithms. These metrics include precision, recall, F1-score, mean Average Precision (mAP), and frames per second (FPS). Based on the experimental results, it concluded that the YOLOv8 algorithm performs better with respect to accuracy and robustness for detecting objects in a complex traffic environment, especially for small and overlapping objects. Though the SSD algorithm performs faster training with a lower computational cost, the accuracy of the object detection results is relatively lower compared to the YOLOv8 algorithm.

**Keywords:** YOLOv8; SSD; Deep Learning; BDD100K; Au-tonomous Driving; Object Detection

## 1. Introduction

Identifying items of the interest and pinpointing their exact locations within pictures or video streams is the core problem of object detection in computer vision [2]. Object detection does both classification and localization at the same time, whereas image classification gives one label to the whole image [2]. Because of these capabilities, it is crucial for real-time intelligent systems like smart surveillance apps, autonomous cars, and traffic monitoring systems [5].

Recent developments in convolutional neural networks and deep learning have greatly increased the precision and effectiveness of object detection models, allowing their application in practical settings [1], [2]. One-stage detectors have drawn a lot of interest among contemporary object detection techniques since they can complete detection in a single forward pass, leading to faster inference times [3], [4].

For applications where real-time performance is crucial, models like You Only Look Once (YOLO) and Single Shot Multi Box Detector (SSD) are frequently used [1]. However, detection accuracy and robustness vary depending on network architecture, feature extraction techniques, and anchor-based versus anchor-free designs. Therefore, it is essential to assess these models in real-world scenarios in order to comprehend their practical advantages and disadvantages.

By precisely identifying cars, pedestrians, traffic signals, and other road players, object detection is essential to main-training road safety in the context of autonomous driving [5]. Detection systems need to operate properly in all types of environmental conditions which include different lighting conditions and various weather patterns and traffic congestion and situations where objects are only partially visible [6].

* Corresponding author: Saurabh Mapari

Because detection mistakes can have a direct impact on system reliability and decision-making processes in safety-critical contexts, these problems make model selection more crucial.

## 2. Literature Review

The field of object detection has advanced through deep learning since its introduction because transformer-based models and convolutional neural networks became available. The traditional approaches to object detection involved sliding windows and hand-crafted features, which lacked any reliability and were computationally expensive [2]. The end-to-end learning paradigm was enabled by deep learning, which significantly improved the accuracy and scalability of object detection for large datasets [1], [4].

Because they can accomplish object localization and classification in one forward pass, one-stage detectors such as YOLO and SSD have been widely used in real-time applecations [1], [5]. To be able to recognize objects of varying sizes effectively, the SSD model introduced multi-scale feature maps with defined anchor boxes [6]. Although SSD can perform fast inference, some studies have shown that SSD may have difficulty recognizing small or heavily occluded objects, especially in complex scenarios such as urban traffic scenes [6].

The YOLO family of models has undergone substantial advancements, with recent approaches concentrating on the efficient fusion of features, improved loss functions, and optimized architectures [1], [2]. With the improved detection precision achieved without compromising the inference speed, recent versions of the model perform well on standard

benchmark datasets [3], [4]. Nevertheless, the applicability of the model on resource-constrained devices could be affected by the rise in the depth and number of parameters of the model, thereby emphasizing the important trade-off between computational complexity and performance [5].

For better knowledge of global features, recent works have also explored transformer-based and hybrid detection architectures, including attention mechanisms [2]. Competitive results have been shown by models such as DETR and its improved versions, particularly in challenging scenarios [2]. However, transformer-based detectors are less appropriate for real-time and embedded systems because they typically require larger training sets and more computational resources [2].

In general, it is found that there is no single object detection model that performs well in all scenarios [5]. Conventional models such as SSD are still applicable for light applications, although advanced YOLO-based detectors provide high accuracy and support real-time processing [6]. The need for unbiased and fair comparative analysis is emphasized by this observation. Based on these findings, the current work investigates the performance trade-offs between YOLOv8 and SSD in autonomous driving scenarios using the BDD100K dataset [1].

### 2.1. Problem Statement

The accuracy, inference speed, and generalization ability of object detection models based on deep learning differ. Thus, selecting an appropriate model for real-time autonomous driving remains a challenging task. The most suitable model for real-time autonomous driving can be identified by comparing YOLOv8 with SSD on a common benchmark dataset.

*Objectives*

- To examine current developments in deep learning-based object identification models and pinpoint important performance patterns.
- To use the BDD100K dataset to implement and train SSD and YOLOv8 models under the same experimental conditions.
- To assess and contrast the two models using common performance measures including inference speed, mean Average Precision (MAP), F1-Score, precision, and recall.
- To evaluate each model's advantages and disadvantages.

## 3. Methodology

### 3.1. Dataset Description

The object detection models are trained and tested on the BDD100K (Berkeley Deep Drive 100K) dataset [5]. The dataset comprises 100,000 high-definition images of driving scenes, along with video frames, captured from real-world driving conditions. This is a large-scale autonomous driving dataset.

For the autonomous driving applications, the collection contains annotated objects including cars, pedestrians, traffic signals, traffic signs, and other road-related things [5]. The BDD100K dataset's coverage of a variety of environmental circumstances, such as day and night scenes, fog, rain, various weather conditions, heavy traffic, and urban roadways, is one of its main features.

The dataset is appropriate for evaluating model performance in practical situations because of its diversity.

For experimental purposes, the dataset was divided into three subsets to enable effective model training and evaluation:

- 70,000 images were used for training
- 10,000 images were used for validation
- 20,000 images were used for testing

This dataset split ensures objective performance evaluation on unseen data while facilitating efficient model learning.

### 3.2. Data Preprocessing

To get the BDD100K dataset ready for efficient YOLOv8 and SSD model training, the data preprocessing was done. Initially, all photos were downsized to fit the models' input specifications; SSD used 300x300 pixel images, while YOLOv8 used 640x640 pixel images [1], [2].

After that, the original dataset annotations were transformed into model-specific formats; SSD employed Pascal VOC (XML) format, whereas YOLOv8 used text-based (TXT) annotation files.

Data augmentation methods such as rotation, random crop-ping, horizontal flipping, and color modifications were used to improve model generalization and lessen overfitting [1], [5].

In order to provide an impartial and equitable assessment of model performance, the dataset was finally divided into training, validation, and testing subsets using a 70:10:20 split.

### 3.3. Model Architecture

**YOLOv8 (You Only Look Once version 8):** Ultralytics created the cutting-edge one-stage object identification model YOLOv8 in 2023 [1]. It is an enhanced version of previous YOLO models (v5–v7) with a number of architectural im-provident that boost overall efficiency, detection speed, and accuracy [1], [2]. The backbone, neck, and detecting head are the three primary parts of the YOLOv8 architecture.

The CSPDarknet53 architecture, which is a variant of the Cross Stage Partial (CSP) model, is the basis for YOLOv8 [2]. Both high-level and low-level features are extracted from input images using the backbone network. The network re-Tains high quality feature information and reduces computing complexity by utilizing bottleneck layers and residual learning [2], [4]. The network generalizes better for objects of different scales due to the improvements brought about by the CSP architecture in terms of gradient flow and learning efficiency. To provide efficient feature fusion, the neck of YOLOv8 combines Feature Pyramid Network and Path Aggregation Network structures [1], [2]. By combining feature maps from various backbone levels, these layers allow for the capture of high-level abstract information. YOLOv8 is useful for detecting tiny, medium, and large items because of this PAN-FPN design, which greatly improves multi-scale object identification [1], [5].

With YOLOv8 anchor-free design, object classes and bounding box coordinates are directly predicted without the need for pre-established anchor boxes [1]. This increases localinaction accuracy and streamlines the training procedure. Non-Maximum Suppression is used during inference to keep the most reliable detections while eliminating redundant bounding boxes [2].

All things considered, YOLOv8 has a number of significant benefits, such as an anchor-free architecture that removes the need for manual anchor tuning, adaptability to support tasks like object detection, instance segmentation, and image classification, and high-speed inference appropriate for real-time applications even on mid-range GPUs [1], [5]. Furthermore, YOLOv8 uses sophisticated data augmentation methods including label smoothing, Mix-up, and Mosaic augmentation to enhance robustness and generalization [1], [2].

**SSD (Single Shot Multibox Detector):** Google Research created the SSD one-stage object detection model in 2016 [2]. In contrast to two-stage detectors like quicker R-CNN, SSD completes object localization and classification in a single forward pass, resulting in quicker inference time without sacrificing competitive detection accuracy [2]. SSD is ideal for real-time object identification applications where low latency is crucial because of this design.

VGG-16, which has been pre-trained on the ImageNet dataset, serves as the foundation of the SSD base network [2]. Convolutional layers are used in place of VGG-16's completely connected layers in SSD to maintain the spatial information required for object detection. This base network serves as the basis for object detection at various scales by extracting rich feature maps from the input images [2].

Conv6, Conv7, and Conv8 are some of the extra convolutional layers that SSD adds to the underlying network to enhance multi-scale detection capability [2]. The spatial resolution of the feature maps created by these layers gradually decreases. While deeper layers record higher-level contextual information for detecting larger items, the earlier layers captrue fine-grained details that are helpful for detecting small things [2].

A predetermined number of default bounding boxes, some-times referred to as anchor boxes, are predicted by the SSD detection head at each feature map position [2]. To accommodate differences in object size, these anchor boxes are defined with various aspect ratios and sizes. The network predicts bounding box offsets, class probabilities, and a confidence score for every anchor box. Non-Maximum Suppression is used during inference to keep the bounding boxes with the greatest confidence scores while removing redundant detections [2].

Taking everything into account, SSD has a number of important advantages, including single-pass detection for fast inference, multi-scale object detection, and an anchor-based framework that enables the model to learn variations in object form and size effectively [6]. SSD can be deployed in realtime applications because of its efficiency in computation, particularly in scenarios with limited computing resources, as in the case of edge devices [5].

### 3.4. Training Configuration

To guarantee an impartial and fair comparison, YOLOv8 and SSD models were trained under the same experimental settings [1], [2]. Python and the PyTorch deep learning framework were used for the training process, while GPU acceleration was used to shorten training times and boost computational effectiveness.

By starting both models with pre-trained weights, transfer learning was used to improve model performance and minimize training effort [1]. By using this method, the models can take use of previously learnt characteristics and converge more quickly.

Adam optimizer was trained with a batch size of 16 and a learning rate of 0.001, depending on the convergence dynamics [2].

### 3.5. Evaluation Metrics

Standard object detection criteria were used to assess the YOLOv8 and SSD models' performance [1], [5]:

Precision: It measures the correctness of detections by determining the ratio of true positives to all of the predicted positives [1], [6].

- Recall: helps to determine the ability of an AI model to identify all objects in an image. [1], [6].
- F1-Score: the harmonic mean of Precision and Recall, balances the performance provided. [6].
- mAP: Mean Average Precision evaluates both classification and localization accuracy [1], [2].
- FPS: Frames Per Second measures inference speed and indicates suitability for real-time applications [5].

## 4. Results and Analysis

The performance of the YOLOv8 and SSD object detection models was evaluated using the BDD100K dataset. Stan-Dard evaluation metrics such as Precision, Recall, F1-Score, and mean Average Precision (mAP) calculated at different Intersection-over-Union (IoU) thresholds were used for performance assessment. These metrics provide a detailed eval-auction of the detection accuracy, robustness, and localization capability of the models.

### 4.1. Quantitative Results

A comparison of the performance model metrics obtained from those two models is presented in Table-I.

**Table 1** Performance Metrics Comparison of Yolov8 and SSD

| Metric | YOLOv8 | SSD |
|---|---|---|
| Precision | 0.604 | 0.520 |
| Recall | 0.345 | 0.310 |
| F1-Score | 0.439 | 0.388 |
| mAP@0.5 | 0.369 | 0.340 |
| mAP@0.5:0.95 | 0.200 | 0.180 |

### 4.2. Performance Comparison Graph

The comparative results of SSD and YOLOv8 on all evaluation metrics are represented in the figure. YOLOv8 consistently outperforms SSD as shown in the bar graph.

The consistent improvement on all metrics reveals the effectiveness of the architectural design of YOLOv8 on difficult object detection tasks.

results indicate that YOLOv8 is more suitable for complex and safety-critical scenarios due to its superior detection accuracy, robustness, and localization capabilities while maintaining real-time processing speed.

SSD has limited capabilities in detecting small and overlapping objects, making it less suitable for complex driving scenarios despite its lower computational complexity and faster training times.

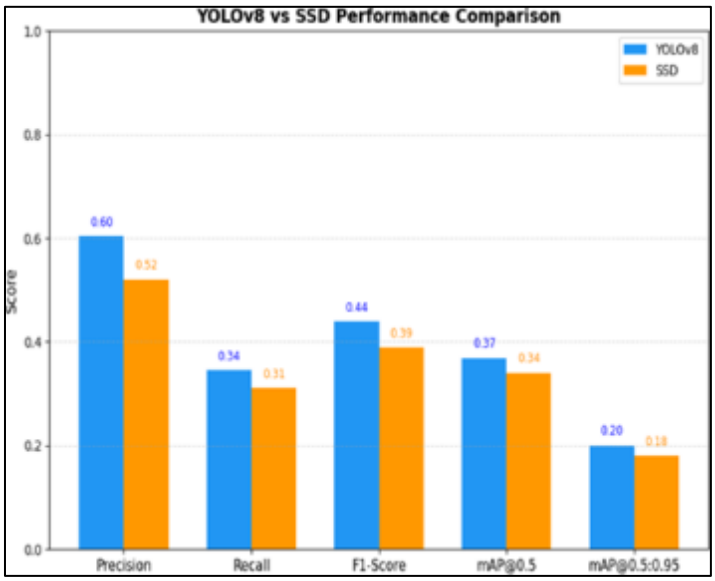


**Figure 1** Performance Comparison of YOLOv8 and SSD

### 4.3. Analysis

On the BDD100K dataset, the experimental results demonstate that YOLOv8 performs better than SSD in terms of detection performance. The accuracy and completeness of the detection results are improved by the increased Precision and Recall ratings, especially in challenging driving scenarios with heavy traffic and occlusion [5], [6].

The PAN-FPN feature fusion step in YOLOv8 enhances its ability to detect small objects, which are common in au-tenuous driving environments [1], [2]. Improved generalization capabilities based on the object size and the environment are also advantages that come with the anchor-free design of YOLOv8 [1], [2].

SSD's detection accuracy is quite low, particularly in com-plex lighting and weather circumstances, despite its faster training and reduced processing complexity [2]. Additionally, YOLOv8 has faster inference speeds in frames per second (FPS), which makes it more appropriate for real-time applications where accuracy and speed are crucial [5].

In general, the results demonstrate that YOLOv8 provides a better trade-off between robustness, real-time processing, and detection accuracy, thus being a more feasible solution for implementation in intelligent transportation systems [5].

## 5. Conclusion

The experiment was conducted using the BDD100K au-tenuous driving dataset to compare and evaluate the performance of YOLOv8 and SSD object identification models. The results indicate that YOLOv8 is more suitable for complex and safety-critical scenarios due to its superior detection accuracy, robustness, and localization capabilities while maintaining real-time processing speed.

SSD has limited capabilities in detecting small and over-lapping objects, making it less suitable for complex driving scenarios despite its lower computational complexity and faster training times

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflict of interest to be disclosed.

## References

[1] M. Sohan, T. Sai Ram, R. Reddy, and C. Venkata, "A Review on YOLOv8 and Its Advancements," in Proc. Int. Conf. Data Intelligence and Cognitive Informatics, Springer, Singapore, pp. 529–545, 2024.

[2] J. Terven, D. C´ordova-Esparza, and J. Romero-Gonz´alez, "A Compre hensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Machine Learning and Knowl edge Extraction, vol. 5, no. 4, pp. 1680–1716, 2023.

[3] Mantri, Ratnamala, and Rais Abdul Hamid Khan. 2025. "YOLO-Based Generalized Framework for Leukemia Cell Detection Using Unified Mi-croscopic Image Datasets". International Research Journal of Multidis-ciplinary Technovation 7 (6):27-40. https://doi.org/10.54392/irjmt2562.

[4] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," in Proc. European Conf. Computer Vision (ECCV), Springer, Cham, pp. 1–21, 2025.

[5] Y. Dai, D. Kim, and K. Lee, "An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8," Electronics, vol. 13, no. 12, Art. no. 2250, 2024.

[6] Q. Su and J. Mu, "Complex Scene Occluded Object Detection with Fusion of Mixed Local Channel Attention and Multi-Detection Layer Anchor-Free Optimization," Automation, vol. 5, no. 2, pp. 176–189, 2024.