**Open Access Library Journal**
2025, Volume 12, e13991
ISSN Online: 2333-9721
ISSN Print: 2333-9705

# Traffic Object Detection Using YOLOv12

**Qizhao Chen**

Graduate School of Information Science, University of Hyogo, Kobe, Japan
Email: af24o008@guh.u-hyogo.ac.jp

**How to cite this paper:** Chen, Q. (2025) Traffic Object Detection Using YOLOv12. *Open Access Library Journal*, **12**: e13991.
https://doi.org/10.4236/oalib.1113991

**Received:** July 21, 2025
**Accepted:** August 11, 2025
**Published:** August 14, 2025



## Abstract

Traffic monitoring plays a vital role in smart city infrastructure, road safety, and urban planning. Traditional detection systems, including earlier deep learning models, often struggle with balancing accuracy, speed, and generalization in diverse and dynamic environments. YOLOv12, the latest model in the YOLO series, introduces architectural improvements such as attention-based mechanisms and efficient layer aggregation, enabling it to overcome limitations related to small object detection, inference latency, and optimization stability. This study evaluates YOLOv12 using a globally sourced traffic dataset that includes varied weather conditions, lighting scenarios, and geographic locations. The model demonstrates strong performance across key object detection metrics, achieving high precision, recall, and mean Average Precision (mAP). Results indicate that YOLOv12 is highly effective for real-time traffic object detection and offers significant improvements over previous approaches, making it a robust solution for large-scale deployment in intelligent transportation systems.

## Subject Areas

Artificial Intelligence

## Keywords

Object Detection, YOLOv12, Traffic Monitoring, Deep Learning, Computer Vision, Real-Time Detection, Bounding Box Localization

## 1. Introduction

In recent years, traffic object detection has become an essential task in the field of computer vision, especially with the growing demand for intelligent transportation systems [1]. Accurate detection of vehicles, pedestrians, and traffic signs enables applications such as real-time traffic monitoring, smart city infrastructure, and autonomous driving [2]. Traditional traffic surveillance systems often relied

on handcrafted features, static thresholds, or motion-based methods [3] [4]. However, such approaches struggle under various environmental conditions such as poor lighting, occlusions, or heavy traffic congestion [4] [5].

With the advancement of deep learning, object detection models have seen significant improvements in both speed and accuracy [4]. Among them, the YOLO (You Only Look Once) family of models has gained wide popularity for its balance between performance and real-time capability. YOLO transforms the object detection problem into a regression task, predicting class labels and bounding boxes in a single forward pass of the network. This makes it highly efficient compared to two-stage detectors such as R-CNN or Faster R-CNN, which first generate object proposals and then perform classification [6].

YOLOv12 is the latest iteration in the YOLO series, featuring an attention-centric architecture specifically designed for real-time object detection. Unlike its CNN-based predecessors, YOLOv12 incorporates a novel Area Attention module to improve the model's receptive field while maintaining low computational cost. It also introduces Residual Efficient Layer Aggregation Networks (R-ELAN) to enhance feature aggregation and improve optimization, especially in large-scale models. Architectural adjustments, such as removing positional encoding, optimizing the MLP ratio, and adopting FlashAttention, enable the model to retain high inference speed. As a result, YOLOv12 achieves state-of-the-art accuracy-latency trade-offs across various model sizes, making it highly suitable for applications like traffic monitoring where both speed and precision are crucial [7].

To evaluate the effectiveness of YOLOv12 in real-world traffic detection scenarios, this study uses a publicly available dataset collected from traffic surveillance cameras in multiple geographic locations. A significant portion of the data comes from Turkish cities such as Bursa, İstanbul, and Konya, while additional images represent traffic scenes from other countries. The dataset captures a variety of environments, including city roads, intersections, and multi-lane streets. It also reflects diverse weather conditions, different times of day, and varying levels of traffic congestion. Each image is annotated with bounding boxes identifying key traffic-related objects, such as different categories of vehicles and pedestrians, making the dataset well-suited for evaluating object detection models in complex and realistic scenarios.

This research aims to investigate how well YOLOv12 can generalize across different geographies and environmental conditions. It explores the model's ability to detect small objects as well as its robustness in scenes with high object density. The model's performance is further evaluated using standard object detection metrics, including precision, recall, and mean Average Precision (mAP).

The main contributions of this paper are as follows. First, the YOLOv12 model is applied to a geographically diverse traffic dataset encompassing a wide range of scenes, lighting conditions, and environmental variations. Second, a comprehensive evaluation of the model's performance is conducted across multiple object classes with comparisons to previous models in the YOLO family. Third, the po-

tential of YOLOv12 for real-time deployment in traffic monitoring systems is explored, along with a discussion of its strengths and areas for future enhancement.

The findings highlight YOLOv12's ability to deliver both high detection accuracy and low inference latency, making it a practical and scalable solution for intelligent traffic applications operating in real-world environments.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 details the methodology. Section 4 presents experiments and results. Section 5 concludes.

## 2. Related Work

This section reviews existing research in two main areas relevant to this study: the evolution of object detection models and the application of these models in traffic detection systems.

### 2.1. Development of Object Detection Models

Object detection has gone through several stages of development, from traditional image processing methods to highly optimized deep learning models. This section summarizes the major developments, focusing on how model design has evolved to improve both accuracy and speed.

#### 2.1.1. Traditional Object Detection

Before deep learning became popular, object detection relied on manual feature extraction and classical machine learning algorithms. Features like Histogram of Oriented Gradients (HOG) [8], Scale-Invariant Feature Transform (SIFT) [9], or Local Binary Patterns (LBP) were extracted from small regions of an image [10]. These features were then passed to classifiers like Support Vector Machines (SVMs) or decision trees.

To search for objects, a sliding window approach [11] was commonly used. The image was scanned at multiple scales and locations, and each window was tested using the trained classifier. Although this method could work for simple scenes, it was computationally expensive and lacked robustness in complex environments with cluttered backgrounds or varying object sizes.

#### 2.1.2. Two-Stage Deep Learning Models: R-CNN Family

The introduction of Convolutional Neural Networks (CNNs) led to a breakthrough in object detection. The Region-based Convolutional Neural Network (R-CNN) [12] was one of the first deep learning-based detectors. It generated region proposals using selective search and then ran a CNN on each proposal to classify it and refine the bounding box. While accurate, R-CNN was very slow due to redundant computations.

Fast R-CNN improved on this by applying the CNN to the entire image only once. Instead of running the CNN on each region proposal, it used a Region of Interest (RoI) pooling layer to extract features for each region from the shared feature map [13]. This reduced computation time and made training faster.

Faster R-CNN further improved the process by introducing the Region Proposal Network (RPN), which learned to generate region proposals directly from the CNN features [14]. This unified the entire pipeline into a single network and significantly reduced inference time. Although Faster R-CNN is still widely used for its high accuracy, especially in scenarios with limited time constraints, it remains slower than single-stage detectors.

#### 2.1.3. One-Stage Models: YOLO, SSD, and RetinaNet

To achieve real-time detection, one-stage detectors were introduced. These models skip the region proposal step and directly predict object classes and bounding boxes from the feature maps.

The YOLO (You Only Look Once) model was the first to frame object detection as a single regression problem. It divides the image into a grid and predicts multiple bounding boxes and class scores for each cell [15]. YOLOv1 was extremely fast but lacked accuracy in detecting small objects.

YOLOv2 [16] and YOLOv3 [17] improved the model's backbone network and introduced multi-scale prediction, allowing the detection of objects of different sizes. YOLOv4 [18] and YOLOv5 further refined the model with techniques like cross-stage partial connections, mosaic augmentation, and efficient training strategies.

YOLOv6 [19] and YOLOv7 [20] emphasized edge deployment and high-speed training, while YOLOv8 [21] introduced anchor-free detection and better post-processing. YOLOv12 [7] extends prior YOLO models by incorporating Area Attention modules to expand the receptive field efficiently and employing Residual Efficient Layer Aggregation Networks (R-ELAN) for enhanced feature aggregation. These additions improve detection accuracy while preserving low inference latency, making the model suitable for real-time applications.

Another popular one-stage model is SSD (Single Shot MultiBox Detector) [22], which also predicts bounding boxes at multiple scales but uses default anchor boxes at each location. SSD is faster than Faster R-CNN but generally less accurate.

RetinaNet [23] introduced the idea of focal loss to handle the class imbalance between object and background. It improved detection accuracy for small and hard-to-detect objects while keeping the inference time low.

#### 2.1.4. Transformer-Based and Hybrid Models

Recently, transformer-based models such as DETR (DEtection TRansformer) [24] have gained attention. DETR reformulates object detection as a direct set prediction problem and uses self-attention mechanisms. While DETR achieves high accuracy and does not rely on non-maximum suppression (NMS), its training time is longer, and inference is slower, making it less suitable for real-time applications.

To overcome these issues, hybrid approaches have been proposed. These models combine CNN backbones with transformer heads or use transformers only in parts of the pipeline. Although these models show promise in general detection

tasks, they often require more computational resources and larger datasets.

In summary, the development of object detection models has shifted from slow and handcrafted methods to highly efficient, end-to-end deep learning architectures. YOLOv12 represents the current frontier in this evolution, combining real-time performance with high detection accuracy across varied environments. Its structure makes it ideal for applications like traffic detection, where both speed and robustness are essential.

Table 1 provides an overview of widely used object detection models, categorized by architecture type and year of introduction, along with their backbones and distinguishing characteristics.

Table 1. Representative object detection models by year.

| Model | Year | Backbone | Remarks |
|---|---|---|---|
| Two-Stage Detectors | | | |
| R-CNN | 2014 | AlexNet/VGG | First CNN-based detector; slow due to selective search and per-region CNN |
| Fast R-CNN | 2015 | VGG-16 | Improves speed with ROI pooling; single-stage training |
| Faster R-CNN | 2015 | ResNet-50/101 + FPN | Introduced Region Proposal Network (RPN) for end-to-end training |
| Mask R-CNN | 2017 | ResNet-50/101 + FPN | Adds mask segmentation branch on top of Faster R-CNN |
| Cascade R-CNN | 2018 | ResNet-101 + FPN | Multi-stage detection head for better localization |
| Libra R-CNN | 2019 | ResNet variants | Balances positive/negative samples and feature fusion |
| One-Stage Detectors | | | |
| YOLOv1 | 2015 | Custom | Real-time detector, unified architecture |
| SSD | 2016 | VGG-16/MobileNet | Multi-scale feature maps for detection at different resolutions |
| YOLOv2 | 2016 | Darknet-19 | Improved accuracy and speed over YOLOv1 |
| RetinaNet | 2017 | ResNet + FPN | Introduced focal loss to handle class imbalance |
| YOLOv3 | 2018 | Darknet-53 | Multi-scale predictions; better accuracy |
| YOLOv4 | 2020 | CSPDarknet53 | Improved backbone and data augmentation |
| YOLOv5 | 2020 | CSPDarknet | Modular, open-source, widely used |
| EfficientDet | 2020 | EfficientNet + BiFPN | Compound scaling for accuracy and efficiency |

**Continued**

| | | | |
|---|---|---|---|
| YOLOv6 | 2022 | EfficientRep | Designed for industrial and edge deployment |
| YOLOv7 | 2022 | E-ELAN | Compound scaling and fast inference |
| YOLOv8 | 2023 | Custom (anchor-free) | Anchor-free; improved post-processing |
| YOLOv9 | 2024 | R-ELAN + Area Attention | Attention-centric, improved receptive field |
| YOLOv10 | 2024 | R-ELAN | Scalable and efficient backbone |
| YOLOv11 | 2024 | R-ELAN + SwiftNet | Lightweight and optimized for real-time detection |
| YOLOv12 | 2025 | R-ELAN + Area + FlashAttention | Advanced attention design; best speed-accuracy trade-off |
| DETR | 2020 | Transformer-based | End-to-end detection; no NMS; slower convergence |
| Deformable DETR | 2021 | Transformer-based | Faster training; better for small objects |

## 2.2. Traffic Object Detection Applications

Object detection has become a core technique in traffic-related computer vision applications. Detecting vehicles, pedestrians, and road signs is fundamental to systems such as automated surveillance, intelligent transportation systems, and autonomous vehicles.

Early traffic detection systems relied on background subtraction, motion tracking, or shape-based classifiers [25]. While useful in simple environments, these methods were sensitive to shadows, occlusion, and camera motion. The adoption of deep learning-based detectors provided a more robust solution.

Faster R-CNN has been widely used in traffic surveillance projects due to its high detection accuracy, especially for vehicles in urban scenes. For example, Gao *et al.* [26] propose an improved Faster R-CNN-based traffic sign detection method that integrates feature pyramid fusion, deformable convolutions, and ROI Align to address challenges such as poor lighting, weather effects, and distant or similar signs. Experiments on the TT100k dataset and real-world vehicle tests show that the proposed method significantly outperforms standard Faster R-CNN and other state-of-the-art approaches, particularly in detecting small or low-visibility signs. Chaudhuri [27] addresses the challenge of vehicle segmentation for smart traffic management under complex conditions such as occlusion, cluttered backgrounds, and variable traffic density. A four-step framework incorporating Faster R-CNN, adaptive background modeling, and extended topological active nets is proposed, achieving superior segmentation accuracy compared to existing methods. Elov *et al.* [28] propose a deep learning framework based on Faster R-CNN for precise vehicle segmentation in smart traffic management, addressing challenges such as

occlusion, lighting variations, and background clutter. By integrating adaptive background modeling and enhanced topological active net deformable models, the method achieves a segmentation accuracy of 98.3%, outperforming existing approaches.

However, the model's two-stage structure limits its suitability for real-time systems like live video feeds or edge devices. To address this, single-stage detectors like YOLO and SSD have been applied in various traffic settings, ranging from highway vehicle counting to pedestrian detection at crosswalks. These models offer a better trade-off between detection accuracy and speed. For example, Zhou *et al.* [29] introduce KCS-YOLO, an enhanced object detection algorithm based on YOLOv5n, designed to improve traffic light recognition for autonomous vehicles under low visibility conditions such as fog, rain, and night-time blur. By incorporating K-means++ clustering, CBAM attention, and a small-target detection layer—alongside dehazing preprocessing—the proposed model achieves a mAP of 98.87%, significantly outperforming baseline methods. Qiu *et al.* [30] present DP-YOLO, an improved traffic sign detection algorithm based on YOLOv8s, designed to enhance small object detection while reducing model complexity. By introducing specialized modules for feature extraction, integrating Transformer-based components, and adopting a new loss function, DP-YOLO significantly reduces parameters (by 77%) and achieves higher detection accuracy across multiple benchmark datasets, making it suitable for edge deployment. Wang *et al.* [31] present an optimized YOLOv5-based framework for traffic sign recognition, incorporating anchor box refinement with k-means++, hyperparameter tuning, and comparative evaluation of YOLOv5 variants to balance accuracy and speed. Experimental results on the CCTSDB dataset demonstrate superior performance over Faster R-CNN and SSD, with mAP reaching 98.1% and real-time inference capability at up to 45 FPS, confirming its effectiveness for intelligent transportation systems under challenging conditions.

For global-scale traffic monitoring, generalization becomes a key issue. Models must handle varying camera angles, lighting conditions, and object densities. Many research efforts now focus on improving robustness through larger and more diverse training datasets, better model regularization, and self-supervised pretraining strategies.

Despite these advancements, achieving both high accuracy and real-time speed remains a challenge in traffic detection. YOLOv12 aims to bridge this gap by offering improved performance while maintaining computational efficiency, making it an attractive option for traffic monitoring systems that operate under real-world constraints.

## 3. Methodology

This section describes the dataset used in this study, the YOLOv12 model setup, training procedures, and the evaluation methods applied to assess model performance. The goal is to ensure that the training process closely reflects real-world

conditions and that the model is fairly evaluated on diverse traffic scenarios.

### 3.1. Dataset

The dataset used in this study is sourced from a public repository on Kaggle, titled Traffic Detection Project[1]. It consists of a rich collection of traffic camera images annotated for object detection tasks. The dataset features images from multiple countries, with a notable concentration in Turkish cities such as Bursa, İstanbul, and Konya. This geographic variety supports the evaluation of detection models across different regional environments, infrastructures, and road designs.

Each image in the dataset is labeled using high-quality bounding boxes that mark the locations of various traffic-related objects. These include vehicles of different types and pedestrians. The annotations follow the YOLO format and are structured to support real-time detection tasks. Table 2 shows the dataset split.

One of the strengths of this dataset is the diversity of environmental conditions captured. It includes images taken in different weather and lighting conditions as well as in complex traffic situations such as busy intersections and multi-lane highways. This variety makes the dataset a strong benchmark for evaluating the robustness and adaptability of object detection models in real-world traffic monitoring scenarios.

Table 2. Dataset split for training, validation, and testing.

| Dataset Split | Number of Images | Percentage |
|---|---|---|
| Training | 7566 | 87% |
| Validation | 805 | 9% |
| Testing | 322 | 4% |
| Total | 8693 | 100% |

### 3.2. Model Setup

YOLOv12 is adopted as the primary object detection architecture for this study. Unlike its predecessors, YOLOv12 introduces an attention-centric design to improve both detection accuracy and computational efficiency. The model incorporates an *Area Attention* mechanism to expand the receptive field with minimal additional cost, which helps capture more contextual information without sacrificing real-time performance. To enhance feature extraction and aggregation, YOLOv12 integrates *Residual Efficient Layer Aggregation Networks* (R-ELAN), which improve gradient flow and optimization stability, particularly in deeper networks.

Further architectural refinements include the removal of positional encodings, the use of a reduced MLP expansion ratio, and the adoption of *FlashAttention* to improve memory and speed efficiency during attention computation. These mod-

[1]https://www.kaggle.com/datasets/yusufberksardoan/traffic-detection-project.

ifications allow YOLOv12 to maintain high detection performance while remaining suitable for deployment in real-time systems.

The model was implemented using the PyTorch-based Ultralytics framework, which provides pre-trained YOLOv12 weights and flexible training pipelines. These pre-trained weights were fine-tuned on the custom traffic dataset to improve performance under the specific conditions represented in these images. The model outputs include bounding box coordinates, class predictions, and confidence scores for each detected object.

### 3.3. Training Procedure

To fine-tune the YOLOv12 model on the dataset, this study used the stochastic gradient descent (SGD) optimizer with a learning rate of 0.01, momentum of 0.9, and a small weight decay of 0.0005 to prevent overfitting. The model was trained for 50 epochs with a batch size of 16 on a single NVIDIA Tesla P100 GPU equipped with 16 GB of VRAM. All input images were resized to 640 × 640 pixels during training.

Data augmentation played a critical role in training. This study applied standard YOLO augmentations such as mosaic augmentation, random horizontal flipping, and HSV color shifting. These transformations helped increase the model's ability to generalize to various lighting and weather conditions. Mosaic augmentation, in particular, was helpful for improving small object detection, which is crucial for recognizing traffic signs and pedestrians from distant views.

During training, model checkpoints were saved after every epoch, and the best-performing checkpoint on the validation set was selected for final evaluation. Loss values were monitored for classification, objectness, and bounding box regression to ensure the model was learning effectively across all aspects of detection.

### 3.4. Evaluation

YOLOv12 is compared with previous versions of the YOLO family to establish its relative effectiveness. The model version of each model is shown in Table 3. Each of these models is designed for real-time object detection with low computational overhead, making them suitable for deployment in edge or latency-sensitive environments.

Table 3. YOLO model versions used in the evaluation.

| Model Name | Version File |
|---|---|
| YOLOv12-nano | yolo12n.pt |
| YOLOv11-nano | yolo11n.pt |
| YOLOv10-nano | yolov10n.pt |
| YOLOv9-tiny | yolov9t.pt |
| YOLOv8-nano | yolov8n.pt |

All models are trained on the same traffic detection dataset under consistent hyperparameter settings and evaluated on the same test split to ensure fair comparison. The evaluation focuses on small and medium-sized objects such as pedestrians, cars, bicycles and motorbikes, which are common in real-world traffic scenes and pose significant challenges for detection accuracy.

To assess the performance of the object detection model, standard evaluation metrics widely adopted in the field are used.

**Precision** measures the proportion of correctly predicted positive detections among all positive predictions. It indicates how accurate the model's predictions are.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

**Recall** quantifies the proportion of correctly detected objects out of all ground truth objects. It reflects the model's ability to find all relevant instances.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

**Mean Average Precision (mAP)** is the primary metric used to evaluate object detectors. It is the average of the average precision (AP) calculated over all object classes and specified Intersection-over-Union (IoU) thresholds.

This study reports two commonly used variants: *mAP*@0.5, which calculates AP at a fixed IoU threshold of 0.5, and *mAP*@0.5:0.95, which averages AP across multiple IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05. The IoU measures the overlap between predicted bounding boxes and ground truth boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Higher mAP values indicate better detection accuracy, considering both localization and classification.

These metrics provide a comprehensive evaluation of the model's effectiveness in both correctly identifying and precisely localizing objects within images.

## 4. Results

Table 4. Performance comparison of YOLOv8 - YOLOv12 on the traffic detection dataset.

| Metric | YOLOv8 | YOLOv9 | YOLOv10 | YOLOv11 | YOLOv12 |
|---|---|---|---|---|---|
| Precision | 0.882 | 0.901 | 0.840 | 0.910 | 0.926 |
| Recall | 0.842 | 0.819 | 0.785 | 0.835 | 0.892 |
| mAP@0.5 | 0.910 | 0.898 | 0.866 | 0.904 | 0.938 |
| mAP@0.5:0.95 | 0.654 | 0.644 | 0.633 | 0.646 | 0.735 |

The evaluation results, summarized in Table 4, indicate that YOLOv12 outperforms earlier YOLO versions across all key object detection metrics on the traffic detection dataset. YOLOv12 achieves the highest precision at 92.6%, demonstrat-

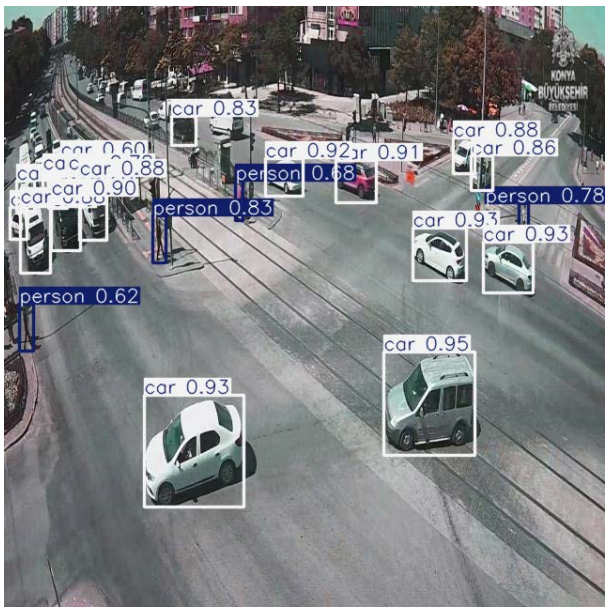
ing its effectiveness in minimizing false positives. It also attains the highest recall of 89.2%, indicating strong capability in detecting most ground truth objects, including small and partially occluded instances.

In terms of overall detection accuracy, YOLOv12 records a mAP@0.5 score of 93.8%, which reflects excellent object classification and localization at a standard overlap threshold. Notably, it also achieves a mAP@0.5:0.95 of 73.5%, significantly outperforming all previous models. This metric is particularly informative, as it averages detection performance across a range of IoU thresholds and better reflects a model's precision in challenging localization tasks. In contrast, the mAP@0.5:0.95 scores of earlier models remain below 66%, with YOLOv8, YOLOv9, YOLOv10, and YOLOv11 scoring 65.4%, 64.4%, 63.3%, and 64.6% respectively.

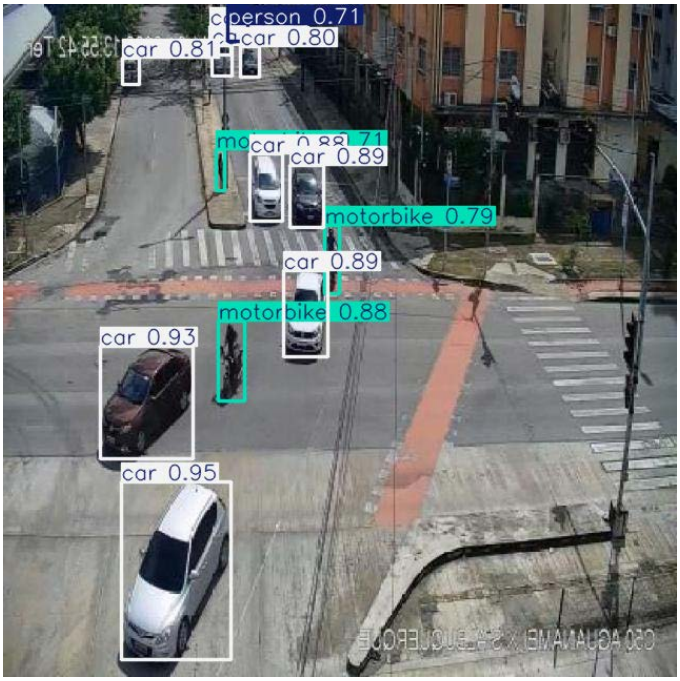
Although YOLOv10 exhibits relatively lower precision and recall, YOLOv9 and YOLOv11 offer incremental improvements over YOLOv8. However, none of these models match the performance gains introduced by YOLOv12, especially in recall and multi-IoU detection performance. The integration of attention mechanisms and enhanced feature aggregation in YOLOv12 contributes to these improvements, particularly in complex urban scenes with small-scale and overlapping objects.

To further illustrate the detection performance of the models beyond numerical metrics, visual predictions are presented using sample images from the test set (**Figure 1**). These examples highlight the ability of YOLOv12 to accurately detect and localize traffic-related objects such as vehicles and pedestrians. These qualitative results complement the quantitative evaluation and offer visual confirmation of the model's robustness and precision in real-world traffic scenes.

Overall, YOLOv12 demonstrates the most balanced and robust detection capability, making it a suitable candidate for real-time traffic monitoring systems. Its




(a) Sample 1




(b) Sample 2

**Figure 1.** Example predictions by YOLOv12 on test images.

consistent performance across both standard and stringent evaluation metrics suggests it is better equipped to handle real-world challenges compared to earlier versions.

## 5. Conclusions

In this study, the YOLOv12 object detection model is applied to a diverse traffic dataset comprising images from multiple countries and varying environmental conditions. The model demonstrated strong performance, achieving high precision and recall, as well as competitive mean Average Precision (mAP) scores. These results confirm YOLOv12's capability to effectively detect various traffic objects, including vehicles, pedestrians, and traffic signs, under real-world scenarios. The balance between detection accuracy and inference speed makes YOLOv12 a practical solution for real-time traffic monitoring systems.

Despite these promising outcomes, certain challenges remain, particularly in detecting small or heavily occluded objects. Future research will explore advanced data augmentation techniques and domain adaptation methods to improve model robustness across diverse traffic environments. Additionally, integrating multimodal data such as video sequences or sensor inputs may further enhance detection accuracy. Finally, investigating hybrid models could offer improvements in capturing complex spatial relationships within traffic scenes.

Overall, this work provides a strong foundation for deploying efficient and accurate traffic detection systems, and the directions outlined offer clear pathways for advancing this research area.

## Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

## References


[1] El-Alami, A., Nadir, Y. and Mansouri, K. (2024) A Review of Object Detection Approaches for Traffic Surveillance Systems. *International Journal of Electrical and Computer Engineering* (*IJECE*), **14**, 5221-5233. https://doi.org/10.11591/ijece.v14i5.pp5221-5233

[2] Khan, A.H., Rizvi, S.T.R. and Dengel, A. (2024) Real-Time Traffic Object Detection for Autonomous Driving. arXiv: 2402.00128. https://arxiv.org/abs/2402.00128

[3] Tarchoun, B., Khalifa, A., Dhifallah, S., Jegham, I. and Mahjou, M. (2020) Hand-crafted Features vs Deep Learning for Pedestrian Detection in Moving Camera. *Traitement du Signal*, **37**, 209-216. https://doi.org/10.18280/ts.370206

[4] Trigka, M. and Dritsas, E. (2025) A Comprehensive Survey of Machine Learning Techniques and Models for Object Detection. *Sensors*, **25**, Article 214. https://doi.org/10.3390/s25010214

[5] Adam, M.A.A. and Tapamo, J.R. (2025) Survey on Image-Based Vehicle Detection Methods. *World Electric Vehicle Journal*, **16**, Article 303. https://doi.org/10.3390/wevj16060303

[6] Jegham, N., Koh, C.Y., Abdelatti, M. and Hendawi, A. (2025) YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions. arXiv: 2411.00201. https://arxiv.org/abs/2411.00201

[7] Tian, Y., Ye, Q. and Doermann, D. (2025) YOLOv12: Attention-Centric Real-Time Object Detectors. arXiv: 2502.12524. https://arxiv.org/abs/2502.12524

[8] Dalal, N. and Triggs, B. (2005) Histograms of Oriented Gradients for Human Detection. 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR*'05), San Diego, 20-25 June 2005, 886-893. https://doi.org/10.1109/cvpr.2005.177

[9] Lowe, D.G. (1999) Object Recognition from Local Scale-Invariant Features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 20-27 September 1999, 1150-1157. https://doi.org/10.1109/iccv.1999.790410

[10] Neha, F., Bhati, D., Shukla, D.K. and Amiruzzaman, M. (2024) From Classical Techniques to Convolution-Based Models: A Review of Object Detection Algorithms. arXiv: 2412.05252. https://arxiv.org/abs/2412.05252

[11] Lampert, C.H., Blaschko, M.B. and Hofmann, T. (2008) Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. 2008 *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, 23-28 June 2008, 1-8. https://doi.org/10.1109/cvpr.2008.4587586

[12] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2016) Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, 142-158. https://doi.org/10.1109/tpami.2015.2437384

[13] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/iccv.2015.169

[14] Ren, S., He, K., Girshick, R. and Sun, J. (2016) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv: 1506.01497.

https://arxiv.org/abs/1506.01497

[15] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/cvpr.2016.91

[16] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/cvpr.2017.690

[17] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. arXiv: 1804.02767. https://arxiv.org/abs/1804.02767

[18] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 2004.10934. https://arxiv.org/abs/2004.10934

[19] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X. and Wei, X. (2022) YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv: 2209.02976. https://arxiv.org/abs/2209.02976

[20] Wang, C., Bochkovskiy, A. and Liao, H.M. (2023) YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, 17-24 June 2023, 7464-7475. https://doi.org/10.1109/cvpr52729.2023.00721

[21] Varghese, R. and M., S. (2024) YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. 2024 *International Conference on Advances in Data Engineering and Intelligent Computing Systems* (*ADICS*), Chennai, 18-19 April 2024, 1-6. https://doi.org/10.1109/adics58448.2024.10533619

[22] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., et al. (2016) SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[23] Lin, T., Goyal, P., Girshick, R., He, K. and Dollar, P. (2017) Focal Loss for Dense Object Detection. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2999-3007. https://doi.org/10.1109/iccv.2017.324

[24] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S. (2020) End-to-End Object Detection with Transformers. In: Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.M., Eds., *Computer Vision—ECCV* 2020., Springer, 213-229. https://doi.org/10.1007/978-3-030-58452-8_13

[25] Hadi, R.A., Sulong, G. and George, L.E. (2014) Vehicle Detection and Tracking Techniques: A Concise Review. *Signal & Image Processing: An International Journal*, **5**, 1-12. https://doi.org/10.5121/sipij.2014.5101

[26] Gao, X., Chen, L., Wang, K., Xiong, X., Wang, H. and Li, Y. (2022) Improved Traffic Sign Detection Algorithm Based on Faster R-CNN. *Applied Sciences*, **12**, Article 8948. https://doi.org/10.3390/app12188948

[27] Chaudhuri, A. (2024) Smart Traffic Management of Vehicles Using Faster R-CNN Based Deep Learning Method. *Scientific Reports*, **14**, Article No. 10357. https://doi.org/10.1038/s41598-024-60596-4

[28] Elov, B., Dauletov, A., Sucharitha, Y., Khalilova, F., Latipova, M. and Abdullayeva, M. (2025) An Intelligent Traffic Management of Vehicles Using Deep Learning Approach in Smart Cities. 2025 3*rd International Conference on Integrated Circuits and Communication Systems* (*ICICACS*), Raichur, 21-22 February 2025, 1-5. https://doi.org/10.1109/icicacs65178.2025.10967703

[29] Zhou, Q., Zhang, D., Liu, H. and He, Y. (2024) KCS-YOLO: An Improved Algorithm for Traffic Light Detection under Low Visibility Conditions. *Machines*, **12**, Article 557. https://doi.org/10.3390/machines12080557

[30] Qiu, J., Zhang, W., Xu, S. and Zhou, H. (2025) DP-YOLO: A Lightweight Traffic Sign Detection Model for Small Object Detection. *Digital Signal Processing*, **165**, Article ID: 105311. https://doi.org/10.1016/j.dsp.2025.105311

[31] Wang, C., Zheng, B. and Li, C. (2025) Efficient Traffic Sign Recognition Using YOLO for Intelligent Transport Systems. *Scientific Reports*, **15**, Article No. 13657. https://doi.org/10.1038/s41598-025-98111-y