Comparative Evaluation of YOLOv5s, YOLOv7-tiny, and YOLOv11s for Real-time Traffic Object Detection

Mete Cem Turan
Computer Engineering
TOBB Economy and Technology University
Ankara, Turkey
metecem.turan@etu.edu.tr

Kerem Elma
Computer Engineering
TOBB Economy and Technology University
Ankara, Turkey
kelma@etu.edu.tr

Kayrahan Toprak Tosun
Computer Engineering
TOBB Economy and Technology University
Ankara, Turkey
ktosun@etu.edu.tr

***Abstract*—Real-time object detection plays a vital role in the advancement of intelligent transportation systems and autonomous driving technologies. This study presents a comparative analysis of three YOLO (You Only Look Once) versions—YOLOv5s, YOLOv7-tiny, and YOLOv11s—trained from scratch using a unified dataset and comparable training settings. Focusing on the traffic object detection problem, the research investigates the architectural evolution and performance improvements of these models across different generations. A balanced dataset is constructed by combining samples from COCO 2017 and the Vehicle Dataset for YOLO, covering three traffic-related classes: car, bus, and motorcycle. Each model is evaluated using standard accuracy and efficiency metrics to reveal trade-offs between detection performance and computational cost. The findings provide insights into YOLO's architectural progression and help identify the most effective version for real-time traffic detection applications.**

***Index Terms*—Real-Time Vehicle Detection, YOLOv5, YOLOv7, YOLOv11, Computer Vision, Model Comparison**

I. Introduction

Real-time object detection has become a fundamental component of intelligent transportation systems and autonomous vehicles. Detecting vehicles and other dynamic traffic elements accurately and efficiently is essential for safe navigation, congestion monitoring, and automated decision-making.

Among numerous detection frameworks, the YOLO (You Only Look Once) family stands out for its real-time inference capability. Over successive versions, YOLO models have improved in accuracy, architectural design, and computational efficiency, making them ideal candidates for deployment in resource-constrained or latency-critical environments. However, most comparative studies of YOLO models rely on pretrained weights or inconsistent datasets, making it difficult to evaluate their true architectural improvements under comparable experimental conditions. Furthermore, limited research focuses specifically on traffic object detection tasks using balanced and unified datasets.

In this study, we train YOLOv5s, YOLOv7-tiny, and YOLOv11s models from scratch using a unified dataset that combines COCO 2017 [1] and the Vehicle Dataset for YOLO [2], focusing on three traffic-related classes: car, bus, and motorcycle. Each model is evaluated under consistent configurations using standard metrics such as mean Average Precision (mAP), Precision, Recall, and Frames Per Second (FPS). The objective is to analyze the architectural evolution and performance trends of YOLO models and to identify the optimal balance between accuracy and efficiency for real-time traffic object detection applications, contributing to safer and more efficient intelligent transportation systems.

II. Related works

Several studies have compared successive YOLO versions to explore their trade-offs in real-time object detection. Olorunshola et al.[3] analyzed YOLOv5 and YOLOv7 on a common benchmark dataset, reporting that YOLOv7 achieved higher mean Average Precision (mAP) and faster inference due to its enhanced Extended Efficient Layer Aggregation Network (E-ELAN) structure. Their findings highlighted the architectural progress between YOLO generations and the performance gains achieved through optimized feature fusion mechanisms.

Building upon this, Dodia and Kumar [4] conducted "A Comparison of YOLO Based Vehicle Detection Algorithms," focusing on vehicle detection and counting for adaptive traffic light systems. Their work implemented YOLOv3, YOLOv5, and YOLOv7 under identical conditions, demonstrating that YOLOv7 achieved the highest detection accuracy of 95.74% with an inference speed of 3.5 ms per frame. They concluded that newer YOLO models, particularly YOLOv7, offer substantial advantages for real-time traffic management and intelligent transportation systems.

Similarly, Gu et al.[5] presented "Comparison and Application of Vehicle Target Detection Methods Based on YOLOv5s and YOLOv7," addressing key challenges in vehicle detection such as false positives, missed detections, and performance degradation in complex environments. Using a combined BIT-Vehicle and custom image dataset, they demonstrated that YOLOv7 significantly outperformed YOLOv5s in both precision and inference speed, especially when detecting small-scale vehicles in cluttered and occluded scenes. Their results

further validated YOLOv7's robustness and efficiency in real-world traffic conditions.

More recently, Das, Ibrahem, and Fouda [6] conducted "A Comprehensive Review on Real-Time Vehicle and Pedestrian Detection Using YOLO," systematically evaluating YOLO versions from v7 through v11. Their findings revealed that YOLOv11 achieved an mAP of 88% at 45 FPS, outperforming all prior versions through innovations such as sparse attention mechanisms and dynamic inference scaling. The study emphasized YOLOv11's scalability and reliability for intelligent transportation and autonomous driving systems.

While these studies collectively demonstrate the rapid evolution and increasing efficiency of YOLO models, most comparisons were conducted on heterogeneous datasets and under differing experimental conditions, which makes direct performance comparison difficult. To address this limitation, the present study performs a unified and controlled evaluation of YOLOv5s, YOLOv7-tiny, and YOLOv11s using a harmonized multi-source dataset derived from COCO and Vehicle Dataset for YOLO. By filtering and combining these sources into a custom dataset focused exclusively on vehicle classes, the data distribution was standardized and label consistency was ensured. This normalization enables a fair and reproducible assessment of detection accuracy, inference speed, and computational efficiency across all three YOLO versions under identical training and validation conditions.

III. Planned Methodology

A. Dataset Construction

The dataset used in this study was systematically constructed through multiple preprocessing and merging stages to ensure balanced and high-quality training data for traffic object detection. Initially, all relevant classes were extracted from the COCO 2017 dataset, and filtered annotation files were generated to include only the required target categories. These filtered instances were then converted from the original COCO format (x_min, y_min, width, height) to the normalized YOLO format (class_id, x_center, y_center, width, height) for compatibility with the YOLO training pipeline.

Due to the limited number of validation samples in the COCO subset, the Vehicle Dataset for YOLO was also processed through the same filtering and conversion steps. The two processed datasets were subsequently merged to increase data diversity and improve class balance. As a result, a domain-specific dataset was created focusing on three traffic-related classes: car (label 0), bus (label 1), and motorcycle (label 2).

The final dataset comprises approximately 17,000 training images and 1,200 validation images, corresponding to about 7% validation data. Within the training set, there are roughly 13,000 car, 4,600 bus, and 4,200 motorcycle instances. This composition provides a realistic yet sufficiently balanced representation of traffic environments, where cars naturally appear more frequently than other vehicle types. The dataset exhibits diverse lighting conditions, camera angles, and urban contexts, allowing the trained models to generalize effectively to real-world scenarios and supporting a fair performance comparison among YOLO versions.

B. Model Configurations

Three YOLO models—YOLOv5s, YOLOv7-tiny, and YOLOv11s—were selected to analyze the architectural evolution of the YOLO family and to benchmark real-time object detection performance under consistent experimental conditions. YOLOv5, released by Ultralytics in 2020, serves as a stable and lightweight PyTorch-based baseline model, offering an excellent trade-off between accuracy and inference speed. It has been widely adopted in both research and industry due to its simplicity, extensibility, and ease of deployment. YOLOv7, introduced in 2022, incorporates Extended Efficient Layer Aggregation Networks (E-ELAN) and model re-parameterization techniques, leading to state-of-the-art real-time detection performance while maintaining computational efficiency. Finally, YOLOv11, the most recent release (2024), introduces major improvements in backbone efficiency, dynamic label assignment, and enhanced feature scaling, providing superior precision and inference speed across multiple benchmarks.

The "small" (s) and "tiny" variants of these models were specifically selected due to their relatively low parameter counts, which make them more suitable for training on locally available GPUs while still preserving the core architectural characteristics of their respective versions. Table I shows the parameter comparison of the selected YOLO variants (YOLOv5s, YOLOv7-tiny, and YOLOv11s), illustrating their compact design and computational efficiency relative to their full-scale counterparts. These three models were chosen not only to represent the key generational milestones of the YOLO family but also because most previous comparative studies have predominantly focused on YOLOv5 and YOLOv7, leaving a research gap regarding how the newest YOLOv11 performs when trained under identical conditions.

All models were trained from scratch (pretrained=False) to ensure a fair comparison of architectural performance without the influence of pretrained weights. The training was conducted with the same hyperparameters: 100 epochs, batch size = 16, input resolution = 640×640, and the Stochastic Gradient Descent (SGD) optimizer (momentum = 0.937, initial learning rate = 0.01). Data augmentation techniques—random horizontal flipping, mosaic augmentation, and brightness/contrast variation—were consistently applied across all models to improve generalization.

Because the dataset contained class imbalance (with car instances being more dominant than bus and motorcycle), class weighting was applied using the –weights parameter to ensure equal contribution of all classes to the loss function. This adjustment mitigated model bias toward the majority class and improved detection consistency across categories.

All experiments will be trained on AMD RX 7800 XT GPUs under identical conditions. The trained models were evaluated using standard object detection metrics, including mAP@0.5,

mAP@0.5:0.95, Precision, Recall, F1-score, Frames per Second (FPS), and average inference time per image, allowing a comprehensive and unbiased performance comparison.

C. Evaluation Metrics

To ensure a comprehensive assessment of detection accuracy, speed, and computational efficiency, several standard metrics were utilized in this study. The mean Average Precision (mAP) was employed as the primary evaluation metric. Specifically, mAP@0.5 measures the average precision across all classes at an Intersection over Union (IoU) threshold of 0.5, while mAP@0.5:0.95 extends this evaluation by averaging precision values over multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05. These two indicators together provide a balanced view of both coarse and fine-grained detection accuracy.

Additionally, Precision, Recall, and F1-score were used to analyze per-class detection quality. Precision quantifies the ratio of correctly identified positive detections to all predicted positives, while Recall measures the ratio of correctly detected positives to all actual positives. The F1-score, defined as the harmonic mean of Precision and Recall, serves as an overall indicator of model reliability, particularly under class imbalance conditions.

To evaluate real-time performance, Frames per Second (FPS) and average inference time per image (ms/img) were measured, reflecting each model's efficiency in practical deployment scenarios. Furthermore, the model size (in megabytes) and GFLOPs (Giga Floating Point Operations) were recorded to assess computational complexity and scalability for real-time applications.

All experiments were conducted under identical training and testing configurations to ensure fairness and reproducibility. Model weights were evaluated on the same validation dataset, and performance comparisons among YOLOv5s, YOLOv7-tiny, and YOLOv11s were based on identical environmental and hardware conditions.

D. Expected Outcome

The primary goal of this study is to provide a comprehensive comparison of YOLOv5s, YOLOv7-tiny, and YOLOv11s in the context of real-time traffic object detection. The expected outcome is the development of a comparative performance chart illustrating the relationship between detection accuracy, inference speed, and computational efficiency across all three models. Through this analysis, the study aims to identify the trade-offs between speed and accuracy, revealing how architectural improvements in successive YOLO versions affect real-time detection when trained under identical conditions.

It is anticipated that YOLOv11s, as the most recent version, will demonstrate superior detection accuracy and faster inference due to its optimized backbone and feature scaling mechanisms. YOLOv7-tiny is expected to achieve a strong balance between precision and speed, while YOLOv5s will serve as a reliable and lightweight baseline for comparative benchmarking.

Ultimately, the results will guide the selection of the most suitable YOLO version for real-time Intelligent Transportation System (ITS) applications, such as vehicle detection and traffic scene understanding, providing insights into how architectural evolution impacts performance in real-world scenarios.

In the implementation phase, the trained models will be integrated into a real-time inference pipeline using OpenCV to evaluate their performance on traffic surveillance videos. The pipeline will measure FPS and inference latency on different hardware settings to validate real-world usability.

TABLE I
COMPARISON OF YOLOV5, YOLOV7, AND YOLOV11 MODELS IN TERMS OF PARAMETER COUNT AND VARIANT COMPLEXITY

YOLO Version	Model Variant	Parameter Count (Millions)
YOLOv5 (2020)	YOLOv5 (overall) YOLOv5s (small)	**7.0–87 M** 7.2 M
YOLOv7 (2022)	YOLOv7 (overall) YOLOv7-tiny	**37–75 M** 6.2 M
YOLOv11 (2024)	YOLOv11 (overall) YOLOv11s (small)	**9.0–258 M** 11.2 M

REFERENCES

[1] https://www.kaggle.com/datasets/awsaf49/coco-2017-dataset

[2] https://www.kaggle.com/datasets/nadinpethiyagoda/vehicle-dataset-for-yolo

[3] Oluwaseyi, Olorunshola Irhebhude, Martins Evwiekpaefe, Abraham. (2023). A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms. Journal of Computing and Social Informatics. 2. 1-12. 10.33736/jcsi.5070.2023.

[4] A. Dodia and S. Kumar, "A Comparison of YOLO Based Vehicle Detection Algorithms," 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/ICAIA57370.2023.10169773.

[5] C. Gu, H. Du, X. Zhang, L. Li, Z. Yang and G. Liu, "Comparison and Application of Vehicle Target Detection Methods Based on YOLOv5s and YOLOv7," 2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2024, pp. 745-749, doi: 10.1109/ICISCAE62304.2024.10761580.

[6] S. Das, M. I. Ibrahem and M. M. Fouda, "A Comprehensive Review on Real-Time Vehicle and Pedestrian Detection Using YOLO," 2025 IEEE 4th International Conference on Computing and Machine Intelligence (ICMI), MI, USA, 2025, pp. 1-7, doi: 10.1109/ICMI65310.2025.11141119.