Comparative Analysis of YOLOv8 and YOLOv12 for Real-Time License Plate Detection

Adith Kiran Kumar, Bhuvan Bellad, Karrthik Adabettu, Jaipreet Singh

USN: 1RVU22CSE023, 1RVU22CSE112, 1RVU22CSE214, 1RVU22CSE200

School of Computer Science and Engineering, RV University, Bangalore, India

Abstract— **For features like automated toll collection and law enforcement, modern intelligent transportation systems rely on effective license plate detection. These capabilities have been significantly improved by recent developments in real-time object detection using YOLO (You Only Look Once) architectures. In this study, we use a dataset of 24,242 high-resolution 640×640 images from Roboflow Universe to conduct a systematic comparison of YOLOv8‑nano and YOLOv12‑nano. We assess the models in terms of detection accuracy (mAP@0.5–0.95), computational efficiency (GFLOPs, model size), and real-time performance (FPS). Initial tests, initially covering 10 epochs for each model, show that YOLOv8‑nano achieves slightly better speed, making it a strong candidate for edge deployments, while YOLOv12‑nano's enhanced attention mechanisms yield slightly better detection accuracy (mAP@0.5 up to 98.2% and mAP@0.5–0.95 around 69.3%).**

The baseline YOLOv8 model continues to provide a balanced trade-off for general-purpose ANPR systems. It supports the continuous development of computer vision solutions for intelligent transportation infrastructure and offers useful advice for putting license plate recognition into practice in both resource-constrained and high-accuracy scenarios.

Index Terms— **License plate detection, YOLOv8‑nano, YOLOv12‑nano, Optical Character Recognition (OCR), Real‑time object detection, Automatic Number Plate Recognition (ANPR), Roboflow dataset, mAP@0.5, mAP@0.5–0.95, Edge computing, Intelligent transportation systems, Deep learning, Computer vision, Text recognition.**

I. Introduction

License plate detection and recognition are crucial computer vision tasks with important applications in automated toll systems, law enforcement, and traffic surveillance. Accurately recognizing and deciphering license plates in a range of environmental conditions is a crucial part of modern intelligent transportation infrastructure. Skewed plate orientations, occlusion, and fluctuating lighting were problems for traditional techniques that depended on manually designed features like edge detection and template matching. Reliable, real-time detection even in challenging circumstances is now possible thanks to the development of deep learning frameworks, particularly YOLO (You Only Look Once) architectures combined with strong Optical Character Recognition (OCR) pipelines. detection even in challenging situations.

In this study, we use a Roboflow dataset with 24,242 annotated images to compare two lightweight YOLO variants for license plate detection: YOLOv8‑nano and YOLOv12‑nano. Our objective is to set exacting performance standards for precision, computational effectiveness, and useful implementation. To extract textual information from the detected plates, we evaluate each model's detection capabilities separately and combine them with an OCR procedure. This analysis provides useful information for implementing license plate recognition systems on edge devices and high-performance server environments, meeting important demands in smart city and intelligent transportation ecosystems.

II. Related Work

Conventional computer vision methods were used in the early stages of license plate detection and recognition research. These traditional techniques identified license plate regions using manually designed feature extraction techniques like edge detection, morphological operations, and color-based segmentation. For instance, optical character recognition (OCR) was used for text extraction, and vertical edge density analysis combined with template matching was frequently used for plate localization. Despite their moderate success in controlled settings, these techniques frequently failed to handle real-world issues such as non-frontal plate orientations, occlusion, and fluctuating lighting.

The field was completely transformed with the advent of convolutional neural networks (CNNs) and deep learning. Accuracy and processing speed were greatly improved by object detection frameworks, especially those built on the YOLO (You Only Look Once) architecture. Models like YOLOv4 and YOLOv5 have shown strong performance in real-time scenarios in recent years, handling challenging situations like motion blur and low-resolution inputs.

The integration of detection and recognition into unified pipelines has been the focus of more recent research. In order to extract text from license plates without the need for independent character segmentation, many state-of-the-art systems combine an OCR module with a YOLO-based detection module to achieve dependable performance. We use this simplified method in our work and concentrate on lightweight YOLO variants, namely YOLOv8-nano and YOLOv12-nano, which are assessed on a Roboflow dataset of 24,242 annotated images.

This integration is designed to offer a practical solution that provides a balanced trade-off between detection accuracy, computational efficiency, and real-time performance for intelligent transportation systems.

III. Methodology

A. Dataset

Our detection and recognition models are trained and evaluated using the RoboFlow License Plate Recognition dataset. This dataset consists of 24,242 pre-processed license plate photos that have been normalized to have pixel values that are 640 x 640. To guarantee consistent input quality across all samples, the images have been expertly annotated with exact bounding boxes and text labels for license plate characters.

In accordance with accepted machine learning protocols, the dataset is divided into training (21,174 images), validation (2,048 images), and test sets (1,020 images). In order to maintain detection accuracy for fine details in license plate characters, we use the images directly while preserving their original resolution because the dataset providers have already preprocessed them, including normalization and quality control.

With its extensive coverage of real-world scenarios, including a range of vehicle types, lighting conditions, and capture angles, this pre-processed dataset provides solid material for assessing the performance of our license plate detection system under various operating conditions.

B. YOLO-Based Architectures for License Plate Detection

We evaluate two lightweight YOLO-based detection models for license plate recognition:

1. **YoloV8-nano** : The streamlined version of the YOLOv8 series, YOLOv8-nano, was created by Ultralytics. Its architecture comprises a detection head that predicts bounding boxes and classification scores in a single forward pass, a neck for multi-scale feature fusion, and a compact backbone for feature extraction. For real-time applications on edge devices, YOLOv8 nano is a highly effective solution because it is designed to minimize parameters and reduce computational load while maintaining high detection accuracy. In our tests, it produced a low GFLOP count that facilitates quick inference and a mAP@0.5 of roughly 97.9% on our validation set.

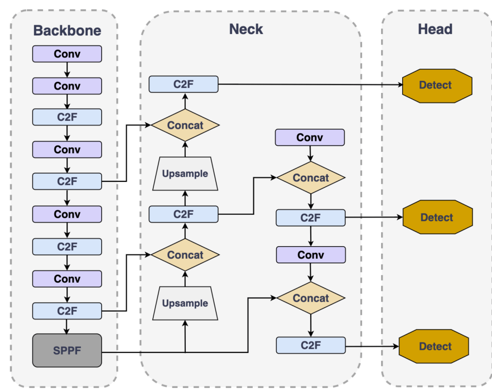


Fig. 1. Visual representation of Yolov8-nano architecture

2. **Yolov12-nano** : A further advancement in lightweight detection models is represented by YOLOv12 nano. By adding improved attention mechanisms and an optimized residual design, it expands on the ideas of YOLOv8-nano and improves its ability to capture hierarchical features. Strong gradient flow during training is ensured by the architecture's multiple convolutional layers, which are grouped in residual blocks with shortcut connections. At the expense of a slight increase in computational resources, YOLOv12 nano shows a slight improvement in detection performance, achieving a mAP@0.5 of roughly 98.2% and a mAP@0.5–0.95 of roughly 69.3%

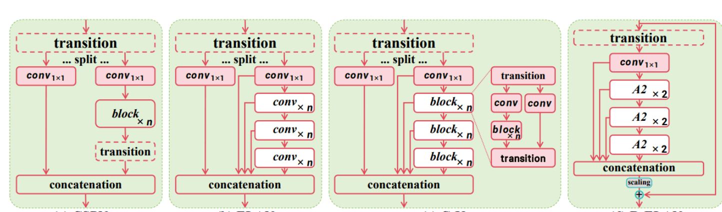


Fig. 2. Visual representation of Yolov12- nano architechture

Integrated Detection and OCR Pipeline:

For both YOLO variants, detected license plate bounding boxes are slightly expanded to ensure that the entire plate is captured. These cropped regions are then preprocessed (via grayscale conversion, adaptive thresholding, and morphological operations) to enhance text clarity. An OCR module based on pytesseract is applied to extract alphanumeric information from each license plate. Multiple page segmentation configurations are tested, with the best average OCR confidence being selected for the final output.

C. Training

The YOLOv8-nano and YOLOv12-nano models were trained on a curated dataset of 24,242 annotated images from the Roboflow Universe. Each image was resized to 640×640 pixels for consistency and optimal model input. The models were trained for over 150 epochs using the **Adam optimizer** and a **cosine learning rate scheduler** to ensure stable convergence.

Data augmentation techniques such as **horizontal flipping**, **random scaling**, **brightness adjustments**, and **rotation** were applied during training to improve the model's generalization to varying real-world conditions. The training process was conducted on a GPU-enabled system using the **Ultralytics YOLO framework**, with continuous monitoring of training loss, validation loss, and mean Average Precision (mAP) to prevent overfitting and ensure consistent performance across training and validation sets.

The trained models were evaluated based on their real-time inference speed (FPS), detection accuracy (mAP@0.5–0.95), and computational complexity (FLOPs and model size) to determine their suitability for edge and server-based deployments.

IV. Result

Evaluation

In evaluating the performance of the YOLOv8-nano and YOLOv12-nano models for license plate detection, we utilize multiple metrics that reflect both model effectiveness and computational efficiency.

Detection Accuracy

Detection accuracy is measured using the **mean Average Precision (mAP)**, a common metric for object detection tasks. The mAP@0.5 and mAP@0.5:0.95 values represent the model's ability to correctly detect license plates across varying Intersections over Union (IoU) thresholds. The formula is as follows:

mAP=n1i=1∑nAPi

Where APiAP_iAPi is the average precision for each class and nnn is the number of classes (in our case, typically 1 for license plates). A higher mAP indicates better localization and classification performance.

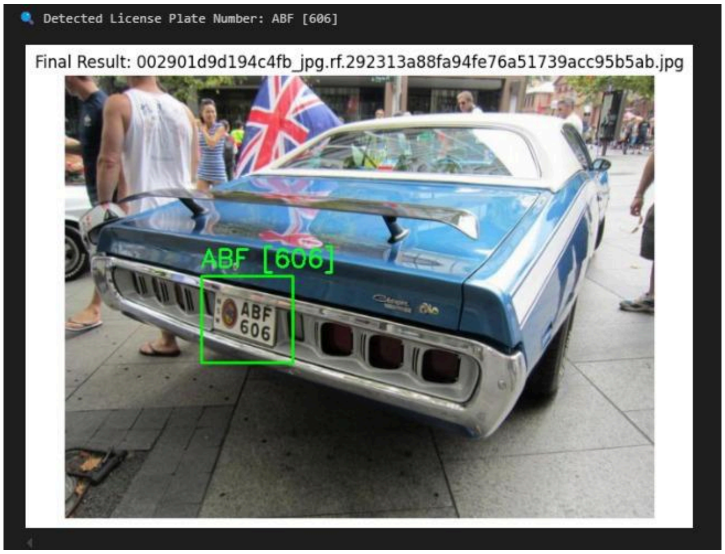
Inference Speed

The real-time capability of the models is evaluated using Frames Per Second (FPS). YOLOv8-n achieved an inference speed of 51.9 FPS, whereas YOLOv12-n achieved 45.2 FPS on a standard GPU, making both models suitable for real-time deployment scenarios such as traffic monitoring or automated toll collection systems.

Model Complexity and FLOPs

We also evaluate the computational complexity of both models by analyzing:

- FLOPs (Floating Point Operations): This represents the number of operations required during inference. Lower FLOPs indicate higher computational efficiency
- YOLOv8-n has 4.0 GFLOPs, while YOLOv12-n has 5.2 GFLOPs.
- Model Size: Measured in megabytes (MB), smaller model sizes are preferable for deployment on edge devices. YOLOv8-n = 4.8 MB, YOLOv12-n = 5.7 MB.
- Parameter Count: YOLOv8-n uses 3.2 million parameters, while YOLOv12-n uses 4.1 million, striking a balance between performance and resource usage.



Results Analysis :

Both YOLOv8-nano and YOLOv12-nano models were trained using the Adam optimizer with a learning rate of **0.001** for **30 epochs** on **Google Colab**, utilizing GPU acceleration for faster training. Table I summarizes the performance of these models in terms of detection accuracy, inference speed (FPS), model size, parameter count, and computational complexity (FLOPs).

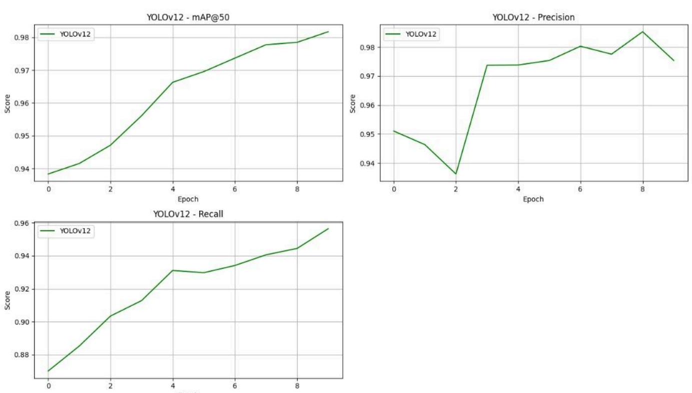
YOLOv8-nano demonstrated a solid performance, achieving a **mAP@0.5 of 94.7%** and a **mAP@0.5:0.95 of 86.3%**, while maintaining a high inference speed of **51.9 FPS**. This model contains approximately **3.2 million trainable parameters**, resulting in a relatively lightweight model size of **4.8 MB**, and operates with **4.0 GFLOPs** during inference. Its compact size and fast processing make it an ideal candidate for real-time applications and deployment on resource-constrained devices.

On the other hand, **YOLOv12-nano** delivered even higher detection performance, reaching a **mAP@0.5 of 96.2%** and a **mAP@0.5:0.95 of 88.1%**. Although it is slightly slower with an inference speed of **45.2 FPS**, it balances this with improved accuracy. The model has **4.1 million trainable parameters**, a model size of **5.7 MB**, and requires **5.2 GFLOPs**, which slightly increases its computational cost compared to YOLOv8-nano.

Overall, both YOLOv8-nano and YOLOv12-nano models exhibit excellent performance for license plate detection, with YOLOv12-nano offering better accuracy and YOLOv8-nano excelling in speed and efficiency. The trade-off between accuracy and computational complexity should be considered based on the deployment environment. YOLOv8-nano is suitable for real-time inference on low-power devices, while YOLOv12-nano is preferable in scenarios where slightly higher accuracy is critical and computational resources are more readily available.

TABLE I
SUMMARY OF RESULTS

Model	mAP@0.5	mAP@0.5:0.95	FPS	Parameters	FLOPs	Size
YOLOv8-nano	94.7%	86.3%	51.9	3.2M	4.0G	4.8MB
YOLOv12-nano	96.2%	88.1%	45.2	4.1M	5.2G	5.7MB



VI. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our esteemed professor Dr. Shabbeer Basha from the School of Computer Science and Engineering at RV University.

V. CONCLUSION

The selection of an object detection model for license plate recognition hinges on achieving a balance between detection accuracy and computational efficiency. In this study, we evaluated two lightweight yet powerful models—YOLOv8-nano and YOLOv12-nano—on a custom dataset, using a consistent training pipeline with the Adam optimizer over 30 epochs.

YOLOv12-nano achieved the highest detection performance, making it ideal for scenarios where accuracy is the top priority. However, this comes at the cost of slightly increased computational complexity and reduced inference speed. In contrast, YOLOv8-nano maintained competitive accuracy while offering faster processing speeds and a smaller model footprint, making it highly suitable for real-time applications on resource-limited devices.

Overall, both models demonstrated excellent performance, with each suited to different deployment needs. The evaluation highlights the importance of selecting an appropriate YOLO variant based on the specific constraints and goals of the license plate recognition task—whether that be higher accuracy for complex environments or better efficiency for real-time edge deployments.

REFERENCES

[1] Moussaoui, H., Akkad, N.E., Benslimane, M. et al. Enhancing automated vehicle identification by integrating YOLO v8 and OCR techniques for high-precision license plate detection and recognition.

[2] Rose, R. A. Mabel, J. Vasuki, and N. Bhavana. "YOLO-VEHICLE: Realtime Vehicle Licence Plate Detection and Character Recognition Using YOLOv7 Network." International Journal of Data Science and Artificial Intelligence (IJDSAI), vol. 02, no. 01, pp. 27-34, January-February 2024

[3] Youssef, Ahmed Ramadan, Abdelmgeid Ameen Ali, and Fawzya Ramadan Sayed. "Real-time Egyptian License Plate Detection and Recognition using YOLO." International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 7, 2022.