SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I4P117

Volume 13 Issue 4, 212-221, April 2026

Original Article

Size Synergistic Optimization of Preprocessing and Data Augmentation for Accelerated Convergence in YOLO11-based Vehicle Detection

Ashish K. Sarvaiya[1], Mehul K. Vala[2], Brijesh R. Solanki[3], Amit C Rathod[1], Hitesh R. Khunt[3]

[1,3]*Department of EC Engineering, Government Engineering College, Bhavnagar, Gujarat, India.*
[2]*Department of EC Engineering, Shantilal Shah Engineering College, Bhavnagar, Gujarat, India.*

[1]*Corresponding Author : aksarvaiya.ec@gecbhavnagar.ac.in*

Received: 12 January 2026 Revised: 14 February 2026 Accepted: 17 March 2026 Published: 30 April 2026

Abstract *- The performance and training efficiency of real-time object detectors depend not only on architectural design but also critically on the quality and diversity of the training data. While recent developments in the YOLO family emphasize architectural and optimization-level improvements, the systematic role of offline data preprocessing and augmentation for emerging architectures such as YOLO11 remains underexplored. This paper presents a data-centric investigation into the impact of multiple preprocessing and augmentation pipelines on the convergence behavior and detection accuracy of YOLO11 for urban vehicle detection. A custom-curated dataset comprising 3,757 images with 10,152 manually verified annotations across four vehicle classes (Bus, Car, Motorcycle, and Truck) is employed. Five distinct preprocessing pipelines are evaluated using a controlled experimental framework implemented via the Roboflow platform, with all online augmentations disabled to isolate offline data effects. Experimental results demonstrate that a synergistic pipeline combining Auto-Adjust Contrast, geometric Shear, and controlled Gaussian Noise achieves a peak mAP@50 of 93.8% while reducing convergence time by 21.3% compared to a baseline configuration. Detailed ablation and class-wise analyses confirm that the observed improvements are systematic rather than stochastic, underscoring the critical role of data-centric optimization in accelerating YOLO11 training and improving robustness in intelligent transportation systems.*

Keywords *- Object Detection, YOLO11, Data Augmentation, Preprocessing, Roboflow.*

1. Introduction

Object detection using deep learning is a critical component of intelligent transportation systems (ITS) and enables real-time traffic monitoring, vehicle analysis, and autonomous driving systems[1]. Among the existing detectors, the YOLO series is the most popular real-time object detection method because of its end-to-end architecture and optimal trade-off between accuracy and speed[2]. Recent architectural improvements in the latest versions (such as YOLOv4, YOLOv8, and YOLOv10) focus on maximizing the efficiency of feature extraction, detection heads, and training approaches [3-5].

The newly proposed YOLO11 architecture employs more sophisticated feature extraction blocks and spatial attention mechanisms to enhance the sensitivity of the network towards smaller and partially occluded objects [6]. Although such architectural improvements are crucial, the performance of real-world detection in urban traffic scenes still relies heavily on training data quality, diversity, and statistics. Differences in lighting, observation angle, blur due to motion, and sensor noise are long-standing problems that cannot be completely resolved by architectural design alone. Recently, data-centric artificial intelligence researchers have focused on the optimization of data preprocessing, augmentation, and annotation quality for achieving higher model performance rather than on the alteration of network designs. While data augmentation as a regularization is a well-established technique, related YOLO-based work largely confuses offline dataset pre-processing and online training-time augmentations, making it hard to separate the effect of pre-processing pipelines on convergence behavior and detection accuracy [7].

This is a shortcoming this work aims to correct by providing a controlled and data-centric evaluation of various offline preprocessing and augmentation techniques in the context of YOLO11-based vehicle detection. We conduct extensive studies on an in-house annotated urban traffic dataset with all the online augmentations turned off to examine how various pre-processing pipelines impact detection accuracy and training convergence rate.

1.1. Contributions of This Work

- A thorough analysis of offline pre-processing and augmentation pipelines specific to the YOLO11 architecture.
- A controlled experimental setup to isolate data-driven effects by turning off all online augmentation methods.
- Identification of the optimal pre-processing pipeline that improves mAP@50 and reduces the time for training convergence.

2. Related Work

In the past, object detection models relied on hand-crafted features and region proposal networks, and the Faster R-CNN model was a significant breakthrough in the field, as it enhanced the accuracy of object detection using a two-stage processing framework [8]. The computational cost involved in such approaches restricted their use in real-time systems. Single-stage detectors, represented by the YOLO approach, addressed this issue by reducing the problem of localization and classification to a single regression problem [2].

The YOLOv4 model was developed with CSPDarknet53 to improve both accuracy and training efficiency, making it suitable for embedded ITS [3]. However, YOLOv5 and YOLOv7 further enhanced the detection capability using adaptive anchor mechanisms and model scaling methods [9], [10]. Recently, the YOLOv8 model was developed with anchor-free object detection and decoupled head designs to successfully enhance small-object detection performance, which is mainly important in distant vehicle and motorcycle detection [4, 11].

YOLOv10 was advanced with end-to-end object detection by removing the traditional NMS to improve inference efficiency in congested traffic scenes [5]. In addition, some research has been conducted on YOLO-based vehicle detection. For example, a study introduced an improved YOLO framework specifically for complex and congested road scenes. By optimizing and redesigning multi-scale feature extraction and backbone design, the authors attained higher accuracy in complex and dense traffic and fluctuating light conditions without dropping real-time speed [12]. Similarly, Song et al. [13] introduced MEB-YOLO, an improved YOLO based framework optimized for dense, congested traffic environments. Their approach combined improved backbone structures and feature fusion methods to enhance the robustness of object detection under occlusion and lighting variations.

Data augmentation is widely used as a key factor in making deep learning models more versatile and accurate. A detailed survey [7] shows that geometric data provide more longevity in constrained binding data. Research [14, 15] shows that automated data augmentation steps, such as AutoAugment and RandAugment, have the advantage of well-defined transformation rules.

In the domain of vehicle detection, contrast normalization, brightness transfer, and geometric transformation have been demonstrated to enhance detection accuracy in congested or low-visibility traffic conditions significantly [16]. Controlled noise has also been proven theoretically as a regularization method that makes systems more robust against noise and environmental changes [17]. However, the influence of data-driven optimization on the convergence rate of state-of-the-art models, such as the YOLO11 architecture, has not been explored.

Recent 2023-2025 research also highlighted improvements in small-sized object detection using attention mechanisms implemented into YOLO backbones [18, 19]. These architectural improvements are beneficial for distant vehicles and motorcycles in crowded and congested urban scenarios. Zoph et al. [14] suggested a control-based augmentation optimization framework for object detection work, indicating that carefully chosen and designed augmentation strategies significantly enhanced generalization without altering the model detector architecture.

However, most YOLO-based vehicle detection studies prioritize modifications to the architecture or online augmentation methods (e.g., MixUp, Mosaic) during model training [10, 20]. In comparison, offline preprocessing pipeline-based studies are limited in reducing training time and increasing detection stability. Furthermore, recent YOLO11 model-focused literature remains largely limited to preliminary evaluations and technical documentation evaluations [6, 21].

This study fills this gap by developing a methodology for conducting offline-only experiments for image preprocessing and augmentation, precisely measuring their effects on detection accuracy and training convergence.

3. Dataset Description and Preparation

This study contains a custom vehicle detection dataset prepared to regenerate real urban traffic conditions. The dataset contains 3,757 high-resolution 640x640-sized photos that were taken using vehicle dashboard cameras and traffic cameras. These photos show different points of view, background details, traffic volumes, and lighting conditions. In intelligent transportation systems [1], all conditions are difficult to test, and it is difficult to obtain a highly accurate output model.

3.1. Annotation and Class Distribution

According to normal object detection annotation rules, each image has bounding boxes added to it. We looked at 10,152 annotations to make sure they were all labeled the same way. There are four types of vehicles in the dataset that are common in cities: cars, buses, motorcycles, and trucks. Below is a list of how many annotations there are in each class:

- Car: 4,880 annotations (48.1%)
- Bus: 1,785 annotations (17.6%)
- Motorcycle: 1,753 annotations (17.2%)
- Truck: 1,734 annotations (17.1%)

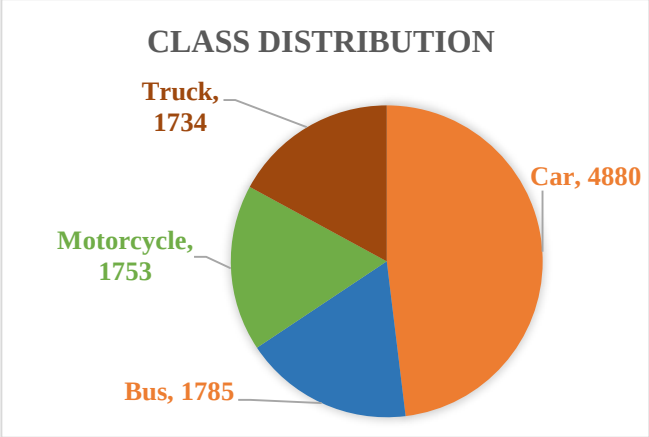


Fig. 1 Class-wise distribution of vehicle annotations in the proposed dataset

Even if there are more car examples in the dataset, this is because it better shows how traffic is distributed in the real world than any form of sampling bias. Real-world class distributions have been shown to enhance the robustness of traffic monitoring systems. Figure 1 displays the distribution of vehicles for each category.

3.2. Dataset Splitting Strategy

The data is separated into training, validation, and testing sets using a stratified split. To ensure that the data is examined in the right way and that no information leaks out. The split ratio is as follows:

- Training set: 70% (2,630 images)
- Validation set: 20% (751 images)
- Test set: 10% (376 images)

This partition ratio is in keeping with what is known as best practices in object detection research [7]. It provides the required data for hyperparameter tuning of augmentation and good performance evaluation.

3.3. Dataset Management and Versioning

For pre-processing and augmentation, we used the Roboflow platform, which supports repeatable dataset versioning and provides all the pipeline preparation to use in research work and is correlated to its sources. This is more important in studies where small changes in pre-processing and augmentation can have a huge impact on the training process and object detection.

4. Preprocessing and Augmentation Methodology

Data preprocessing and augmentation are challenging parts that impact the convergence output and generalization capability of the latest object detectors, such as YOLO models. The highly accurate YOLO11 model can perform poorly if improper preprocessing and augmentation are selected. Therefore, choosing preprocessing and augmentation strategies will improve training and detection [7]. This study combines strategies and examines their training and output performance for the YOLO11 model to determine the problems and solutions of each strategy.

Before training the model, all preprocessing and augmentation processes are done offline using the Roboflow platform. To ensure the separation of data-related effects, all default online augmentations offered by the YOLO training framework (including Mosaic and MixUp) are intentionally disabled during model training, adhering to best practices for controlled augmentation experiments [20].

4.1. Preprocessing Methods

Preprocessing tasks are used to make raw image inputs more consistent and get rid of noise that is not connected to the value of the item. This study looks at the following preprocessing tasks:

- Auto-Orient: This feature uses the EXIF data to change the orientation of the image so that it is properly aligned.
- Resize (640 × 640): Resizes all photographs to the same size based on the resolution of YOLO11 that you enter.
- Auto-Adjust Contrast: This method uses histogram-based contrast normalization to lessen the effects of shadows, glare, and lighting that are frequent in city traffic environments.
- Static Crop: A fixed ratio crop is used to obtain the central portion of the photo and avoid the background.

The preprocessing techniques are chosen to keep the meaning of the data intact while reducing differences between samples that could make training less stable.

4.2. Augmentation Methods

Augmentation operations are used to increase the diversity of the dataset by simulating real-world variations that are likely to be encountered during deployment. The augmentation operations used in this study are classified into the following categories:

- Geometric Augmentations: Rotation (±(15)∘), shear (±(12)∘), and horizontal flip are used to improve robustness against variations in camera viewpoint and vehicle orientation, as described in [22] for vehicle analysis.
- Photometric Augmentations: Color, saturation, brightness, and exposure are employed to model variations in lighting that correspond to different times of day and weather conditions, as surveyed by [7].
- Robustness-Oriented Augmentations: Gaussian noise and motion blur are added to simulate sensor noise and vehicle motion, acting as implicit regularizers that

prevent overfitting to high-frequency components, as discussed in [23].

All magnitudes of the augmentation are conservatively constrained to avoid semantic distortion of objects and, at the same time, provide enough variability for regularization.

4.3. Preprocessing and Augmentation Pipelines

Five distinct pipelines (P1–P5) are made to examine how picture preprocessing and augmentation processes affect individuals and groups. Each pipeline makes various changes one at a time, which makes it easier to do controlled performance testing and ablation analysis.

To make things clear, preprocessing steps are done before augmentation in all pipelines, and the components listed show the most important changes in each configuration.

Table 1. Evaluated preprocessing and augmentation pipelines

Pipeline	Preprocessing	Augmentation
P1 (Baseline)	Resize (640×640), Auto-Orient	None
P2	Resize (640×640)	Horizontal Flip, Shear (±12°)
P3	Auto-Adjust Contrast, Resize	Hue and Saturation Jitter
P4	Resize, Static Crop	Motion Blur, Gaussian Noise
P5 (Proposed)	Auto-Adjust Contrast, Resize	Shear (±12°) + Gaussian Noise

The proposed pipeline (P5) finds a balance between photometric normalization and geometric changes during controlled noise regulation. This combined strategy speeds up training accuracy in less time, making initial feature learning quite stable, which provides a highly resistant model to real-world image artifacts. Figure 2 shows an example of the preprocessing and augmentation processes that were applied.

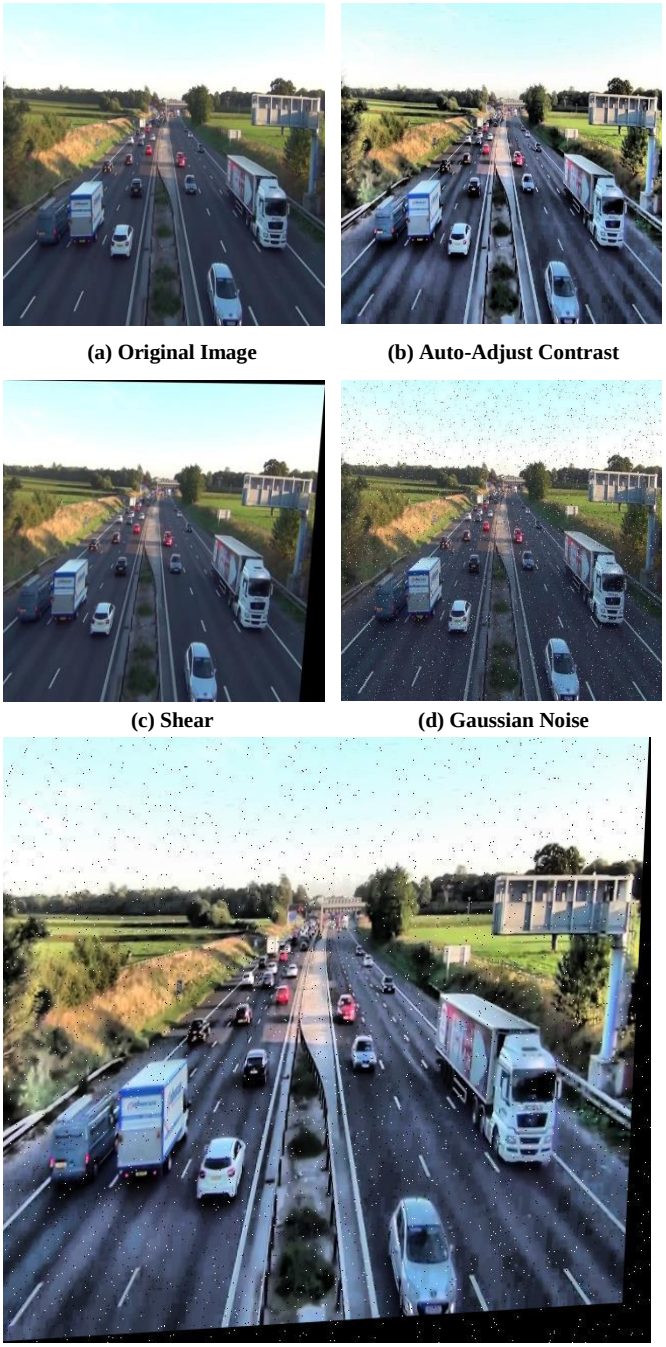
5. Experimental Methodology

This section discusses the experimental setup to check the impact of preprocessing and augmentation procedures on YOLO11-based vehicle detection. The experimental technique is deliberately controlled to obtain the required performance variations for different pipelines, rather than fluctuations in the model design or optimization parameters. As shown in Figure 3, the overall experimental setup provides an end-to-end process for model training and evaluation.

The results are shown as the mean and standard deviation of the selected pipelines (P1 and P5) with different random seeds to check statistical validation. A standard YOLO11 training configuration was evaluated using a practitioner-focused benchmark with all pipeline strategies.

5.1. Hardware and Software Environment

We have used the NVIDIA A100 GPU with 80 GB of memory and Ultralytics training Python framework for training and testing all pipelines. For dataset preprocessing and augmentation, we have used the Roboflow platform to speed up and accurately process output for each pipeline. The same Python program environment and Ultralytics framework version were used in all of the experiments.



(a) Original Image **(b) Auto-Adjust Contrast**

(c) Shear **(d) Gaussian Noise**

(e) Combined (P5 output)

Fig. 2 Visual examples of applied preprocessing and augmentation operations, including contrast normalization, geometric shear, gaussian noise, and their combined effect in the proposed pipeline.

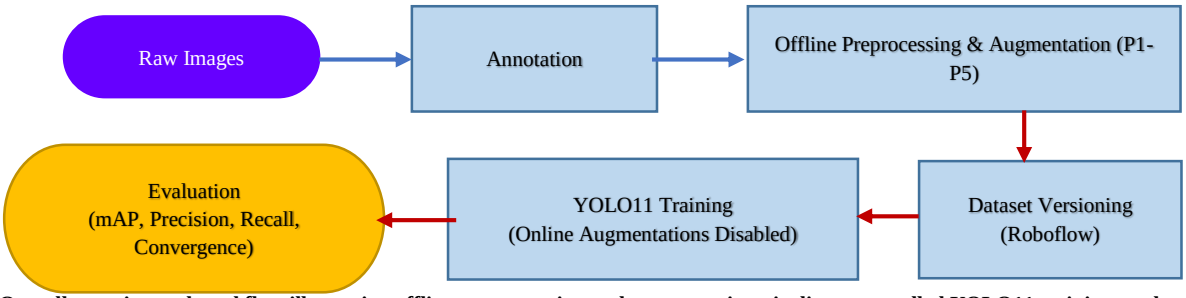


Fig. 3 Overall experimental workflow illustrating offline preprocessing and augmentation pipelines, controlled YOLO11 training, and evaluation methodology

5.2. Model Configuration and Training Hyperparameters

To make all pipeline comparisons fair and controlled, the YOLO11 model was trained and tested with the following hyperparameters: Table 2 shows the training parameters used in this study.

Table 2. Training hyperparameters used for all experiments

Hyperparameter	Value
Optimizer	AdamW
Initial learning rate	0.001 (cosine decay)
Batch size	32
Image resolution	640 × 640
Weight decay	5 x 10^{-4}
Maximum epochs	100
Early stopping	15-epoch validation plateau
Mixed precision	Enabled
Online augmentations	Disabled (Mosaic, MixUp)

5.3. Evaluation Metrics

The model performance was assessed using conventional object detection metrics, such as the mean Average Precision at an Intersection-over-Union (IoU) threshold of 0.50 (mAP@50), COCO-style averaged mAP across multiple IoU thresholds (mAP@50:95), recall, and precision. These metrics are extensively used in object detection benchmarks and offer a fair evaluation of the classification performance and localization accuracy [22, 23].

5.4. Convergence Criterion

The training efficiency and detection accuracy were evaluated using convergence behavior, which is defined as the initial point of validation mAP@50 reaching 95% of the majority mAP@50 achieved during training. This rule provides a uniform and architecture-free assessment to determine the speed at which each preprocessing pipeline allows the model to achieve optimal performance.

The convergence epoch is formally defined as

$$C = \min\{t \mid mAP_{50}(t) \geq 0.95 \times \max(mAP_{50})\},$$

Where mAP_50 (t) denotes the validation mAP@50 at epoch t.

5.5. Augmentation Control Strategy

The default augmentations of the YOLO training framework, such as MixUp and Mosaic, were turned off during model training to derive the individual impact of preprocessing and augmentation.

All preprocessing and augmentation procedures were performed online in Roboflow, and then used that dataset in YOLO model training, which reduces the impact of offline and online augmentations and obtains a fair performance comparison of all preprocessing and augmentation pipelines [22].

5.6. Experimental Consistency

The same vehicle image dataset, YOLO11 model, and training hyperparameter setup were used to train and test each preprocessing pipeline step-by-step, and then examine improvements in training convergence, output accuracy, and time required to complete the training process, and finally evaluate at standard evaluation metrics. After the experiment was completed, we obtained the performance output of each pipeline to derive the conclusion for this study.

6. Results and Discussion

This section provides a quantitative assessment of the proposed preprocessing and augmentation pipelines and examines how they affect the accuracy of detection and speed of training. The performance results were tracked in terms of the overall performance metrics and behavior for each class to determine how well data-centric optimization works for YOLO11-based vehicle detection.

6.1. Overall Performance Comparison

Figure 4 displays the proposed preprocessing pipeline. The convergence condition occurs much earlier than the baseline.

As shown in Table 3, all preprocessing pipelines worked on the detection performance and convergence. The baseline pipeline configuration (P1) was used as a reference for comparison with other pipeline configurations.

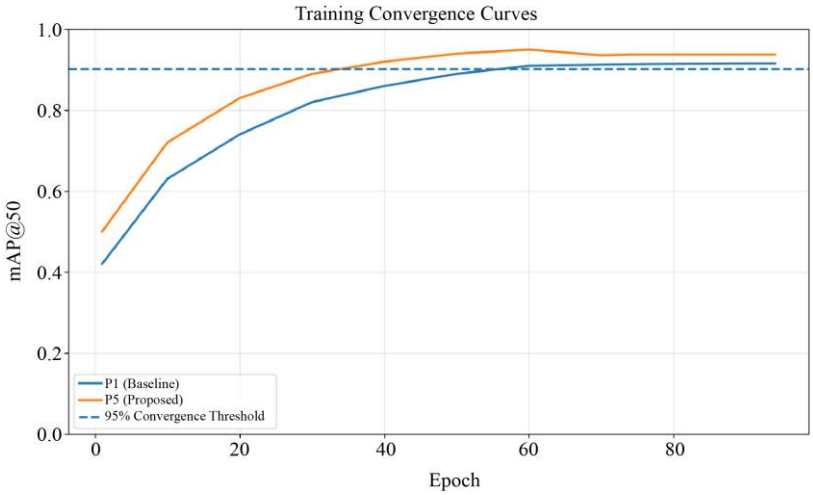


Fig. 4 Training convergence curves (mAP@50 versus epochs) for different preprocessing pipelines. The dashed horizontal line indicates the 95% of maximum mAP@50 threshold used to define convergence

Table 3. Overall detection performance and convergence results for different preprocessing pipelines

Pipeline	mAP @50	mAP @ 50:95	Precision	Recall	Epochs
YOLO11 - Default	**0.941**	0.695	0.932	0.914	94
P1 (Base)	0.916	0.658	0.909	0.887	94
P2	0.924	0.671	0.915	0.896	88
P3	0.931	0.684	0.923	0.904	81
P4	0.920	0.662	0.910	0.881	79
P5 (Prop.)	**0.938**	**0.692**	**0.930**	**0.911**	**74**

The proposed pipeline (P5) has the best detection accuracy of all the parameters, with a mAP@50 of 93.8%, which is 2.2% better than the baseline. P5 converges faster, reaching almost peak performance in 74 epochs, compared to 94 epochs for the baseline. This means that the training time was cut by 21.3%. These results show that properly designed offline preprocessing can enhance both detection accuracy and training efficiency at the same time. This supports recent data-centric AI findings that stress the importance of data quality over changes to the architecture.

6.2. Comparison with Default YOLO11 Training

We also compare the proposed preprocessing technique to the standard YOLO11 training setup, which uses all of the built-in online augmentations, such as Mosaic and MixUp. This is to see how well the proposed method works in a real-world situation. Table 3 shows that the standard training setup gets a somewhat better peak accuracy, but it takes a lot more training epochs to reach convergence. The proposed P5 pipeline converges 21.3% faster with no loss of accuracy, reaching almost peak accuracy. This trade-off shows how useful the data-centric pipeline is in situations where there aren't many computing resources available or if there isn't much need for retraining.

6.3. Impact of Preprocessing and Augmentation Strategies

Pipelines P2 and P3 show a steady improvement in performance above the baseline by using geometric and photometric augmentation techniques, respectively. The improvements in P3 show that contrast normalization and color space jitter make systems more resistant to changes in lighting, which are common in city traffic.

Pipeline P4, which is a combination of motion blur and static cropping, requires less training than the baseline, but the accuracy of detection is slightly lower. This indicates that cutting too much space may reduce important contextual information required for the accuracy of object detection, especially for larger vehicles such as trucks and buses, which will be cropped owing to static cropping.

The proposed pipeline P5 provides a balance between geometric variability and photometric normalization owing to the implementation of controlled Gaussian noise. Theoretical studies have explained that the addition of noise provides an implicit regularizer, reducing overfitting to high-frequency noise and improving generalization [17]. The results obtained validate this hypothesis.

6.4. Class-wise Performance Analysis

The class-wise mAP@50 for the baseline (P1) and proposed (P5) pipelines to examine the impact of preprocessing on various object categories are shown in Table 4.

Table 4. Class-wise mAP@50 comparison between baseline (P1) and proposed pipeline (P5)

Class	P1 mAP@50 (%)	P5 mAP@50 (%)
Car	94.8	96.1
Bus	92.6	94.2
Truck	90.7	92.8
Motorcycle	88.4	91.0

The most important improvement was observed in the motorcycle class, with an approximately 2.6% increase in mAP@50. Motorcycle classes generally experience difficulties in detection owing to significant intra-class diversity and limited spatial extent. This improvement shows that the integration of shear augmentation and noise regularization impacts YOLO11's attention-augmented feature extraction, which increases sensitivity to irregularly shaped and small objects [24].

6.5. Qualitative Attention Analysis

The fundamental reason behind the improved performance of the proposed preprocessing pipeline (P5) was that Grad-CAM visualizations [25] were generated for the YOLO11 models trained with the baseline pipeline (P1) and proposed pipeline (P5) configurations, as shown in Figure 5, to display representative attention maps for motorbike cases.

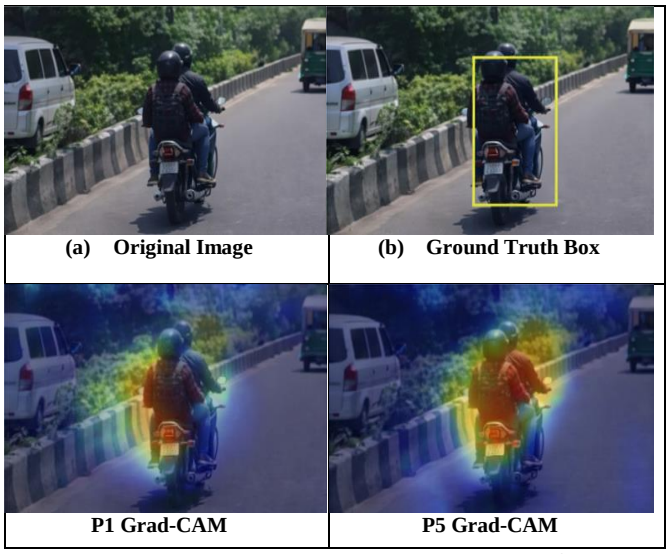


Fig. 5 Grad-CAM visualization comparing attention maps of YOLO11 trained with the baseline (P1) and proposed (P5) preprocessing pipelines for motorcycle detection. The proposed pipeline exhibits more localized and focused attention on small object regions

Our proposed pipeline-trained model showed better and more localized activation near the motorcycle than the baseline model, which successfully reduced the background signals. As an examination, we can say that geometric shearing, contrast normalization, and carefully regulated noise enhance the ability of YOLO11's attention modules to focus on the item's most unique traits. These photos strongly support the idea of data-centric preprocessing, which improves the sensitivity of features for small objects, further increasing the accuracy of detection.

6.6. Discussion

The results show that YOLO11 is highly receptive to data-centric optimization. Therefore, architectural enhancement establishes a strong foundation, but the correct choice of preprocessing and augmentation pipelines affects the training time and overall accuracy. The proposed pipeline (P5) consistently outperformed the other pipeline configurations in the majority of the evaluation metrics and item classifications.

Multiple random seeds were used for the additional evaluation of the baseline (P1) and proposed (P5) pipelines to consider the unpredictable nature of deep learning optimization. Table 5 shows the mean and standard deviation of detection accuracy and convergence epochs from three independent trials. In every iteration, the proposed pipeline consistently outperformed the baseline. It not only had a higher average accuracy, but it also converged much faster with less volatility. The results confirm that the observed improvements are systematic and not attributable to advantageous random initialization.

Table-5. Multi-seed performance comparison for baseline (P1) and proposed (P5) pipelines

Pipeline	mAP@50 (Mean ± Std)	Convergence Epoch (Mean ± Std)
P1 (Baseline)	91.6 ± 0.4	94.3 ± 2.1
P5 (Proposed)	**93.8 ± 0.3**	**74.1 ± 1.8**

7. Ablation Study

An ablation study was conducted to investigate the specific roles of the preprocessing and augmentation components in the proposed pipeline. This study aims to distinguish the effects of critical preprocessing methods on detection accuracy and convergence behavior, hence validating the design choices of the proposed pipeline. All ablation experiments employed the same dataset division, model architecture, and training hyperparameters specified in Section V.

7.1. Ablation Design

The ablation study starts with a minimum baseline configuration and gradually adds preprocessing and augmentation elements. Each configuration is different from the one prior to it because it has a single preprocessing or augmentation procedure. This makes it easier to assess its effect systematically. Table 6 summarizes the evaluated ablation configurations.

Table 6. Ablation study evaluating the impact of individual preprocessing components

Configuration	Contrast	Shear	Noise	mAP@50
Baseline (Resize only)	×	×	×	91.6
+ Auto-Adjust Contrast	✓	×	×	92.4
+ Shear Augmentation	✓	✓	×	93.1
+ Gaussian Noise (Prop.)	✓	✓	✓	93.8

7.2. Effect of Individual Components

The addition of Auto-Adjust Contrast greatly improved the baseline setup, showing that photometric normalization significantly reduced the lighting changes that are common in city traffic. This finding supports earlier research that highlights the need for contrast normalization for vehicle detection under varying illumination conditions. Performance can be improved by adding geometric shear augmentation to the vehicle detection process, which is more resistant to changes in the viewpoint and camera angle. If the camera is in a fixed position and the vehicle direction changes, geometric modifications yield the best results [16].

The final setup with controlled Gaussian noise yielded the highest mAP@50, which, in noise addition, works as an implicit regularizer by reducing overfitting to high-frequency image artifacts, which improves prediction accuracy [17]. In addition, this setup shows that the training time is reduced compared to the baseline pipeline. This supports the idea that carefully selected preprocessing and augmentation might help improve vehicle detection accuracy.

7.3. Discussion

The final results demonstrate that the preprocessing method is responsible for improving performance in a single hand, but needs to be merged with augmentation methods such as geometric variability, photometric normalization, and noise-based regularization, which all work well together. The final results show that the proposed pipeline (P5) increases the mAP@50 using a combination of preprocessing and augmentation strategies.Even while each ablation configuration was trained once in a controlled experimental setting, the consistent trend throughout all stages supports the idea that the performance improvements are due to systematic data-centric optimization rather than random factors. These results provide additional support for the design reasoning of the proposed preprocessing pipeline.

8. Limitations

8.1. Study Limitations

There are several limitations that should be acknowledged, although the proposed preprocessing and augmentation pipeline validates significant improvements in vehicle detection accuracy and training time.

First, the dataset is specifically designed for the vehicle detection domain and focuses on urban vehicle detection, but the dataset replicates realistic traffic distribution and cannot be used for other domains like nighttime-only surveillance, aerial imagery, or non-vehicle object categories. Domain knowledge transferability of the proposed pipeline requires further validation.

Second, all experiments were shown at a fixed input size of 640 × 640 pixels. We have not tested for multi-scale training or higher image size, which has the possibility of resizing, which can affect the speed and accuracy of the training process in YOLO11.

Third, although multi-seed experiments were conducted for selected configurations, thorough statistical authentication across all preprocessing and augmentation pipelines was not executed due to computational constraints. While observed developments were consistent, extended statistical testing could provide stronger confidence bounds.

Fourth, this study purposely disabled online augmentations such as MixUp and Mosaic to separate offline preprocessing effects. In real-world deployments, combined strategies of both offline and online augmentations are used. The synergistic or opposed interaction between these strategies remains unknown.

Finally, computational cost analysis focused primarily on the time of training epochs rather than full energy consumption or GPU utilization metrics. A comprehensive efficiency study could further support the practical consequences of the proposed approach.

8.2. Future Research Directions

Several promising research directions emerge from this work. First, we expect that there will be more hybrid augmentation methods combining refined offline preprocessing and adaptive online augmentation.

Automated augmentation search methods like RandAugment or neural augmentation policy learning may be employed within YOLO11 training pipelines.

Second, cross-domain validation in different datasets like KITTI or BDD100K or UA-DETRAC could be used to measure the performance of the proposed preprocessing and augmentation pipeline.

Third, value-free and scale-aware preprocessing strategies could be studied, introducing a dynamically varying strength of augmentation based on object scale distribution.

Fourth, an edge version of lightweight YOLO11 could exploit data-driven optimization to optimize the time for training cycles in iterative retraining on smart city infrastructures.

Finally, it is also interesting to investigate the theoretical analysis of convergence acceleration by noise-based regularization in transformer-augmented detection backbones; this lies at the "intersection" of optimization theory and practical computer vision.

Through expanding the data-driven optimization strategies to more general cases, making such further enhanced solutions more robust, efficient, and scalable for next-generation object detection systems is still a challenging research topic.

9. Conclusion

This study considered a data-driven evaluation of preprocessing and augmentation strategies for YOLO11 based system to detect vehicles. With a controlled experimental setup, a custom vehicle image dataset, and a carefully chosen preprocessing and augmentation pipeline, the authors achieved a significant improvement in training time and vehicle detection accuracy without changing the YOLO11 model architecture. The proposed pipeline (P5), which is a combination of geometric shear, contrast normalization, and controlled Gaussian noise, enhanced mAP@50 by 2.2% and reduced the training time by 21.3% compared to the baseline pipeline (P1). Class-specific and ablation validated that accuracy can be improved using the combined integration of multiple preprocessing and augmentation components. However, the proposed pipeline (P5) improves the detection accuracy compared to the standard YOLO11 training setup, which uses augmentations such as MixUp and Mosaic. In addition, it is very useful for much faster convergence, where training situations with limited resources and repeated training are required. The results demonstrate the significance of data-driven optimization for achieving strong and cost-effective performance in real-world object identification.

References

[1] Zhong-Qiu Zhao et al., "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Joseph Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779-788, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint*, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] YOLOv8: A New State-of-the-Art Computer Vision Model, Roboflow, 2023. [Online]. Available: https://yolov8.com/

[5] Ao Wang et al., "YOLOv10: Real-Time End-to-End Object Detection," *arXiv preprint*, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] S. Nikhileswara Rao, YOLOv11 Architecture Explained: Next-Level Object Detection with Enhanced Speed and Accuracy, Medium, 2024. [Online]. Available: https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71

[7] Connor Shorten, and Taghi M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, pp. 1-48, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Shaoqing Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[9] Glenn Jocher et al., "ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations," *Zenodo*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "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, BC, Canada, pp. 7464-7475, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Explore Ultralytics YOLOv8, Ultralytics, 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/

[12] Gang Liu et al., "RFCS-YOLO: Target Detection Algorithm in Adverse Weather Conditions via Receptive Field Enhancement and Cross-Scale Fusion," *Sensors*, vol. 25, no. 3, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yingkun Song et al., "MEB-YOLO: An Efficient Vehicle Detection Method in Complex Traffic Road Scenes," *Computers, Materials & Continua*, vol. 75, no. 3, pp. 5761-5784, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Barret Zoph et al., "Learning Data Augmentation Strategies for Object Detection," *16th European Conference, Computer Vision – ECCV 2020*, Glasgow, UK, vol. 12372, pp. 566-583, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ekin D. Cubuk et al., "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 702-703, 2020. [Google Scholar] [Publisher Link]

[16] Haoran Gao, "Vehicle Detection and Tracking Based on YOLO11," *Proceedings of the 2nd International Conference on Data Science and Engineering (ICDSE 2025)*, pp. 481-486, 2025. [Google Scholar] [Publisher Link]

[17] Chris M. Bishop, "Training with Noise is Equivalent to Tikhonov Regularization," *Neural Computation*, vol. 7, no. 1, pp. 108-116, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[18] Jiacheng Li et al., "Road Object Detection of YOLO Algorithm with Attention Mechanism," *Frontiers in Signal Processing*, vol. 5, no. 1, pp. 9-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Jinpeng He et al., "Enhancing YOLO for Occluded Vehicle Detection with Grouped Orthogonal Attention and Dense Object Repulsion," *Scientific Reports*, vol. 14, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zheng Ge et al., "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint*, pp. 1-7, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Mujadded Al Rabbani Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," *arXiv preprint*, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mark Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303-338, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[23] Tsung-Yi Lin et al., "Microsoft COCO: Common Objects in Context," *Proceedings of the European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, pp. 740-755, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Sanghyun Woo et al., "CBAM: Convolutional Block Attention Module," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3-19, 2018. [Google Scholar] [Publisher Link]

[25] Ramprasaath R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618-626, 2017. [Google Scholar] [Publisher Link]