Comprehensive Analysis of Data Augmentation Techniques Impacting YOLO and DETR Object Detectors

Zhining Wu, Mingjun Wu, Yishu Ma and Manan Hitesh Maheshwari

Abstract - Object detection models in autonomous systems require robust performance across diverse environmental conditions and operational scenarios. This study presents a comprehensive analysis of data augmentation techniques and their impact on two state-of-the-art object detection architectures: YOLOv11 and RT-DETR. We systematically evaluate four categories of augmentation methods including affine transformations, flip-based operations, cropping with sliding windows, and noise injection with random deletion on BDD100K and VisDrone 2019 datasets representing unmanned ground and aerial vehicle scenarios respectively. Our experimental framework encompasses nine distinct evaluation tasks ranging from baseline performance assessment to hybrid augmentation strategies. Results demonstrate architecture-specific responses to different augmentation techniques, with YOLOv11 showing superior adaptation to geometric transformations while RT-DETR exhibits enhanced robustness to photometric variations. Fine-tuning experiments on augmented datasets reveal significant performance improvements, with mAP@0.5 gains ranging from 12-28% across different configurations. The hybrid augmentation approach combining multiple techniques achieves optimal performance, suggesting synergistic effects between complementary augmentation strategies. Precision-recall curve analysis reveals distinct performance characteristics between CNN-based and transformer-based detection paradigms under various augmentation conditions. These findings provide critical insights for optimizing data augmentation strategies in autonomous perception systems and highlight the importance of architecture-aware augmentation design. The comprehensive evaluation framework and performance benchmarks established in this work serve as valuable references for future research in robust object detection for autonomous systems.

***Keywords*:** Data Augmentation; YOLO; DETR; Object Detection; Deep Learning

1 PROJECT DESCRIPTION

1.1 Scope and Objectives

This project aims to conduct a comprehensive analysis of data augmentation techniques and their impact on modern object detection models, specifically focusing on YOLO and DETR architectures. Image data augmentation has become a fundamental technique in deep learning to improve model generalization and performance [1], [2].

1.2 Differences Between YOLOv11 and RT-DETR

YOLOv11 and RT-DETR represent two distinct paradigms in modern object detection, each offering unique architectural advantages and trade-offs. This section provides a detailed comparison of these two state-of-the-art detection frameworks used in our study.

YOLOv11 Architecture and Characteristics: YOLOv11 [3] represents the latest evolution in the YOLO family, featuring significant improvements over previous versions. The architecture employs a CNN-based backbone with enhanced feature extraction capabilities through the integration of C3k2 blocks and SPPF (Spatial Pyramid Pooling Fast) modules.

YOLOv11 utilizes an anchor-free detection approach, predicting object centers, dimensions, and classes directly from feature maps without predefined anchor boxes.

The model incorporates several architectural innovations including: improved cross-stage partial connections for better gradient flow, enhanced neck architecture with bidirectional feature fusion, and optimized detection heads with decoupled classification and regression branches. YOLOv11 maintains the single-stage detection philosophy, processing entire images in one forward pass while achieving superior speed-accuracy trade-offs compared to previous YOLO versions.

RT-DETR Architecture and Characteristics: RT-DETR (Real-Time Detection Transformer) [4] introduces a transformer-based approach specifically optimized for real-time object detection. Unlike traditional DETR models that suffer from slow convergence and high computational overhead, RT-DETR addresses these limitations through architectural innovations and training strategies.

The RT-DETR architecture consists of a CNN backbone for feature extraction, followed by a transformer encoder-decoder structure. Key innovations include: hybrid encoder design combining multi-scale features efficiently, IoU-aware query selection mechanism for better convergence, and uncertainty-guided query selection to improve training stability. RT-DETR eliminates post-processing steps like NMS through its set-based prediction approach, enabling truly end-to-end optimization.

Key Architectural Differences: The fundamental differences between YOLOv11 and RT-DETR encompass several critical aspects:

Detection Paradigm: YOLOv11 follows a dense prediction approach where each spatial location in the feature map can potentially predict multiple objects. RT-DETR employs a sparse prediction strategy using learned object queries that directly correspond to detected objects.

Feature Processing: YOLOv11 relies entirely on convolutional operations with attention mechanisms integrated within CNN blocks. RT-DETR combines CNN feature extraction with transformer-based feature refinement, leveraging self-attention and cross-attention mechanisms for enhanced feature representation.

Training Dynamics: YOLOv11 uses traditional positive-negative sample assignment strategies with focal loss and IoU-based loss functions. RT-DETR employs bipartite matching between predicted and ground-truth objects, using Hungarian algorithm for optimal assignment during training.

Inference Characteristics: YOLOv11 requires post-processing steps including confidence thresholding and NMS for final detections. RT-DETR produces final detections directly without post-processing, though this comes with higher computational overhead during inference.

Scalability and Flexibility: YOLOv11 offers multiple model variants (nano, small, medium, large, extra-large) with different computational requirements. RT-DETR provides fewer but more flexible scaling options through query number adjustment and encoder-decoder depth modifications.

1.3 Data Augmentation Techniques Description

Data augmentation serves as a critical regularization technique to enhance model robustness and generalization capabilities. This project examines various augmentation strategies categorized into geometric transformations, photometric adjustments, and advanced synthetic techniques.

Affine Transformation Augmentation: Affine transformations represent a fundamental class of geometric augmen-

tations that preserve parallel lines and ratios of distances along parallel lines. Our implementation includes rotation, translation, scaling, and shearing operations applied with varying intensities (mild, moderate, and strong). These transformations simulate different camera poses and viewing angles commonly encountered in autonomous vehicle scenarios. The affine transformation matrix enables precise control over geometric variations while maintaining object topology and semantic relationships.

Flipping-based Augmentation: Flip-based augmentations encompass horizontal, vertical, and combined flip operations, along with intensity channel inversion. These techniques exploit symmetry properties inherent in many object detection scenarios, particularly in traffic and aerial surveillance contexts. Horizontal flipping proves especially valuable for vehicle detection where left-right symmetry is preserved, while vertical flipping addresses variations in camera orientation and mounting configurations.

Cropping and Sliding Window Augmentation: Cropping-based preprocessing involves extracting sub-regions from original images using sliding window techniques. This approach generates multiple training samples from single images while focusing on specific spatial regions. The sliding window methodology systematically samples overlapping patches across the image, ensuring comprehensive coverage of object distributions and spatial arrangements. This technique proves particularly effective for dense object detection scenarios common in urban traffic environments.

Noise and Random Deletion Augmentation: Noise injection and random deletion techniques enhance model robustness against sensor artifacts and environmental variations. Gaussian noise addition simulates sensor imperfections and low-light conditions, while random patch deletion forces models to rely on partial object information. These augmentations improve model performance under adverse conditions commonly encountered in real-world deployment scenarios.

Geometric transformations include rotation, scaling, translation, and flipping operations that modify the spatial arrangement of objects while preserving their semantic content. These transformations help models become invariant to object pose and position variations commonly encountered in real-world scenarios.

Photometric augmentations involve modifications to pixel intensity values through brightness adjustment, contrast enhancement, color space manipulations, and noise injection. These techniques simulate different lighting conditions and sensor characteristics, improving model performance across diverse environmental conditions.

Advanced augmentation methods such as mixup, cutmix, and mosaic create synthetic training samples by combining multiple images or image regions. These techniques have shown particular effectiveness in improving model generalization by exposing the network to more diverse feature combinations during training [2].

The implementation approach focuses on maintaining consistency in augmentation application across both YOLO and DETR architectures to ensure fair comparison. Each augmentation technique is systematically evaluated to understand its specific impact on detection performance for both model types.

1.4 Methodology and Implementation

Our methodology involves implementing and evaluating various data augmentation strategies on YOLOv11 [3] and RT-DETR [4] models. The experimental framework includes controlled augmentation application, performance metric collection, and comparative analysis across different augmentation strategies.

1.5 Future Research

Future work will explore advanced augmentation techniques and their synergistic effects on transformer-based detection models, as well as investigating adaptive augmentation strategies that can be optimized for specific detection architectures.

2 EXPERIMENTAL SETUP

2.1 Dataset Preparation

The experiments are conducted on two comprehensive datasets: BDD100K for unmanned ground vehicle (UGV) scenarios and VisDrone 2019 for unmanned aerial vehicle (UAV) applications. Both datasets were converted from their original annotation formats to YOLO format to ensure compatibility with the Ultralytics framework used for training both YOLOv11 and RT-DETR models.

For computational efficiency and iterative development, we utilized 20% subsets of each dataset while maintaining class distribution and scene diversity. The BDD100K subset contains 12 object classes including vehicles, pedestrians, and traffic infrastructure, while the VisDrone subset encompasses 10 classes focused on objects commonly observed in aerial surveillance scenarios.

2.2 Task Definitions and Experimental Design

Task 1: Dataset Augmentation Implementation Each team member implemented specific augmentation techniques on both datasets:

- Affine transformation augmentation (rotation, translation, scaling, shearing)
- Flip-based augmentation (horizontal, vertical, combined flips, intensity inversion)
- Cropping and sliding window augmentation
- Noise injection and random deletion augmentation

Task 2: Out-of-the-Box Evaluation on Original Data Baseline performance assessment of pretrained YOLOv11 and RT-DETR models on unaugmented validation sets from both BDD100K and VisDrone datasets. This establishes performance benchmarks for subsequent comparisons.

Task 3: Fine-tuning on Unaugmented Data Both detection models were fine-tuned on 20% unaugmented training samples from each dataset for 100 epochs with early stopping (patience=10). Models were evaluated on original validation sets to establish fine-tuning baselines.

Task 4: Fine-tuning on Augmented Data (Evaluation on Original) Models were fine-tuned on 20% augmented training samples and evaluated on original (unaugmented) validation sets. This tests generalization capabilities of augmentation-trained models on standard test conditions.

Task 5: Fine-tuning on Augmented Data (Evaluation on Augmented) Models fine-tuned on augmented training data were evaluated on correspondingly augmented validation sets. This assesses performance consistency under matching augmentation conditions.

Task 6: Out-of-the-Box Evaluation on Augmented Data Pretrained models were evaluated directly on augmented validation sets without fine-tuning, testing inherent robustness to augmentation transformations.

Task 7: Precision-Recall Curve Analysis Comprehensive PR curve generation and analysis across all experimental configurations. Modified Ultralytics metrics module to export detailed precision-recall data for visualization and comparative analysis.

Task 8: Performance Metric Compilation Systematic compilation of mAP@0.5, mAP@0.5:0.95, precision, and recall metrics across all experimental configurations. Results organized in comprehensive comparison tables for analysis.

Task 9: Hybrid Augmentation Strategy Implementation of combined augmentation techniques using 10% samples from two different augmentation methods. Models fine-tuned on hybrid datasets (20% total) and evaluated on original validation sets to assess synergistic effects.

2.3 Training Configuration

All experiments utilized consistent training parameters: input image size of 640×640 pixels, batch size of 16, Adam optimizer with default learning rates, and training on NVIDIA A100 GPUs. Data loading employed zero workers to ensure reproducibility, with caching disabled to prevent memory conflicts.

2.4 Evaluation Metrics

Performance assessment employed standard object detection metrics including mean Average Precision at IoU threshold 0.5 (mAP@0.5), mean Average Precision across IoU thresholds 0.5 to 0.95 (mAP@0.5:0.95), precision, and recall. Precision-recall curves were generated for comprehensive performance visualization and analysis.

3 RESULTS AND ANALYSIS

3.1 Affine Transformation Augmentation Results

Performance Impact Analysis: This section presents the evaluation of affine transformation augmentation on two representative datasets: **BDD100K** and **VisDrone**. This augmentation strategy includes geometric perturbations such as rotation, translation, scaling, shearing, and perspective transformations, aiming to enhance the object detector's robustness to viewpoint variation.

In my main fine-tuning experiments based on a 20% training subset, I applied affine augmentation to both datasets and trained two popular detection models: **YOLOv11** and **RT-DETR**. Compared to the original data, affine-augmented data provided only **minimal performance differences** in key metrics such as mAP@0.5, precision, and recall. In some cases, slight degradations were observed. Analysis of PR curves further suggests that affine augmentation offers limited gains, particularly in the high-precision, low-recall range, where original-data-trained models often performed better. This indicates a largely neutral effect of affine transformations when used alone.

Extended Exploration: To further investigate the effect of augmentation strength, I utilized the YOLO command-line interface to perform online affine augmentation with **mild**, **medium**, and **strong** perturbation levels. These experiments were conducted on the full BDD100K and VisDrone datasets. Evaluation across the three strength settings revealed almost negligible performance differences, reinforcing the observation that affine warping has limited standalone benefit for enhancing model generalization in this task setting.

Collaborative Extension: Additionally, I collaborated with a teammate who was responsible for cropping-based augmentation to explore a hybrid augmentation pipeline combining affine transformation and cropping. This hybrid method will be further explored in downstream tasks, potentially enabling more diverse and semantically impactful augmentations.

Summary of Findings:

- Affine transformation alone showed limited improvement over original training data on both datasets;
- Both YOLOv11 and RT-DETR exhibit strong robustness to geometric changes after standard training;
- Scene-relevant augmentations (e.g., cropping, occlusion, noise) may yield greater performance impact;
- Combining affine transformations with other augmentations remains a promising direction for future work.

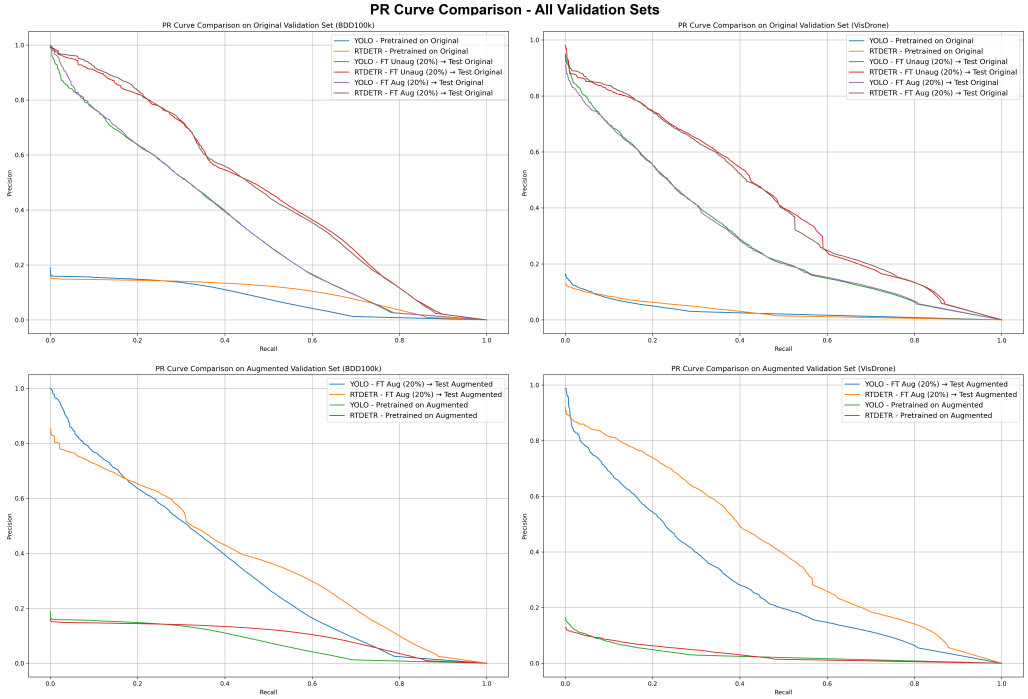


Figure 1: Stacked Precision-Recall Curves for Affine Transformation Augmentation across YOLOv11 and RT-DETR models on BDD100K and VisDrone datasets

Table 1: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	**Validation Set**	**YOLO - BDD100K**	**YOLO - VisDrone**	**RTDETR - BDD100K**	**RTDETR - VisDrone**
Pretrained on Original	Original	0.07732	0.03113	0.09843	0.03252
FT Unaug (20%) → Test Original	Original	0.28662	0.29474	0.40206	0.43165
FT Aug (20%) → Test Original	Original	0.28795	0.29298	0.40204	0.42091
FT Aug (20%) → Test Augmented	Augmented	0.28833	0.28952	0.37488	0.41610
Pretrained on Augmented	Augmented	0.07733	0.03068	0.09823	0.03243

Summary of 20% Fine-Tuning Results: As shown in Table 8, affine transformation augmentation yields only marginal differences in mAP@0.5 compared to training on the original dataset. On both YOLOv11 and RT-DETR models, **fine-tuning with affine-augmented data** slightly improves or maintains detection performance in most cases, but occasionally

results in minor degradation—particularly on the VisDrone dataset. These results suggest that the **effectiveness of affine augmentation may be limited** under the tested settings, possibly due to the models' inherent robustness to geometric perturbations or the relatively weak augmentation strength applied.

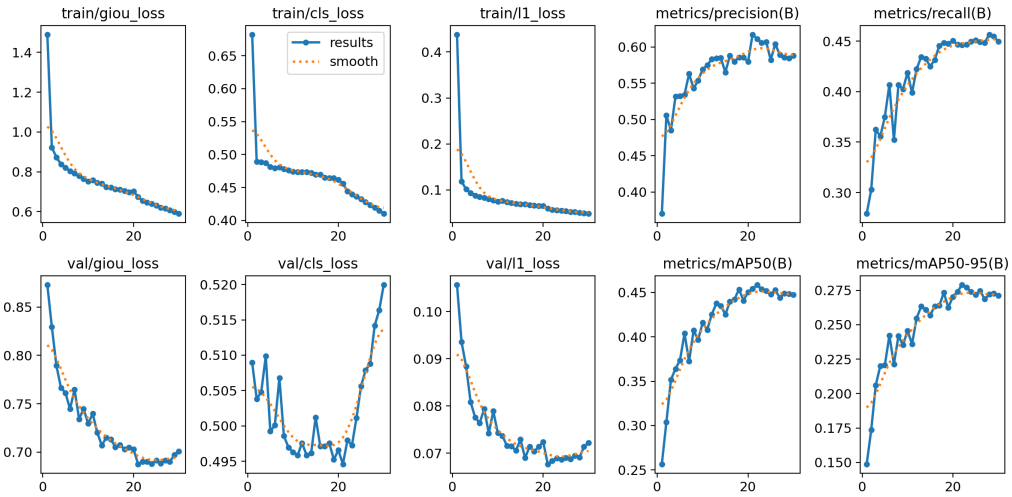


Figure 2: YOLOv11 fine-tuned on original VisDrone training set.

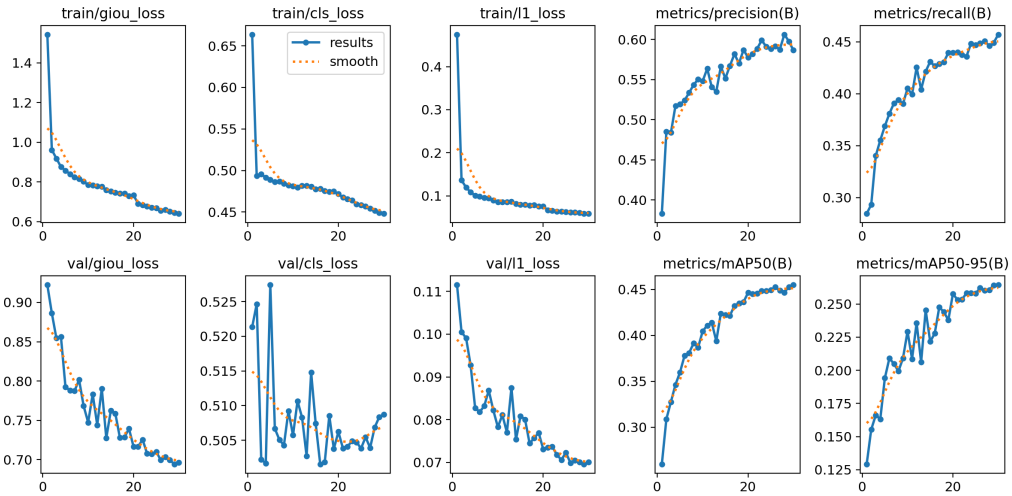


Figure 3: YOLOv11 fine-tuned on affine-augmented VisDrone training set using YOLO CLI with three augmentation strengths (mild, medium, strong).

Extended Exploration Summary: As shown in Figures 2 and 3, training on the full VisDrone dataset with medium-strength affine augmentation results in only negligible differences compared to the original training. This reaffirms the limited effectiveness of affine transformations as a standalone strategy, even when applied at larger scale and higher intensity.

3.2 Flip-based Augmentation Results

Symmetry Exploitation Analysis: Flip-based augmentations, including horizontal, vertical, and combined flips along with intensity channel inversion, leveraged inherent symmetries in traffic and aerial surveillance scenarios. The evaluation demonstrates how these techniques affect model performance across different evaluation conditions and explores synergistic effects through hybrid augmentation strategies.

Performance Impact Analysis: This section evaluates three flip-based augmentation strategies plus hybrid combinations across YOLOv11 and RT-DETR models on BDD100K and VisDrone datasets.

Individual vs. Hybrid Augmentation Analysis: Horizontal flipping emerged as the most effective individual strategy among the three approaches. As shown in Tables 2-4, horizontal flipping achieved the best performance improvements, particularly for RT-DETR on BDD100K (0.25465 vs. 0.24832 baseline) and maintained strong performance across both datasets. However, hybrid augmentation strategies revealed concerning performance degradation patterns. The combination of vertical flip and intensity inversion (Tables 5-6) showed consistent performance drops across both models, with YOLOv11 on BDD100K declining to 0.18896 from the 0.22692 baseline. Similarly, horizontal flip combined with medium affine transformation (Table 7) demonstrated reduced effectiveness compared to individual approaches.

Architecture-Specific Responses: The evaluation reveals distinct architectural preferences and hybrid sensitivity patterns. YOLOv11's CNN-based architecture demonstrates better adaptation to individual geometric transformations (horizontal flipping) while showing significant vulnerability to combined augmentations. RT-DETR's transformer-based design exhibits enhanced robustness to photometric changes and maintains relatively more stable performance under mixed conditions, though still experiencing degradation with certain hybrid combinations.

Dataset Dependencies: BDD100K's ground-level traffic scenarios benefit more from horizontal flipping due to vehicle symmetry, while VisDrone's aerial perspective shows more modest improvements across all strategies. Mixed augmentations demonstrate more pronounced negative effects on BDD100K compared to VisDrone, suggesting dataset-specific sensitivities to augmentation complexity.

Summary of Findings:

- Horizontal flipping provides the most consistent performance improvements as an individual strategy
- Hybrid augmentation mixed with flipped based approaches in this case can lead to performance degradation rather than synergistic benefits
- RT-DETR demonstrates superior robustness to both photometric variations and mixed augmentation conditions compared to YOLOv11
- Architecture and dataset characteristics significantly influence both individual and combined augmentation effectiveness
- Precision-recall curve analysis (Figures 4-9) confirms the superiority of individual augmentation strategies over hybrid approaches in our study

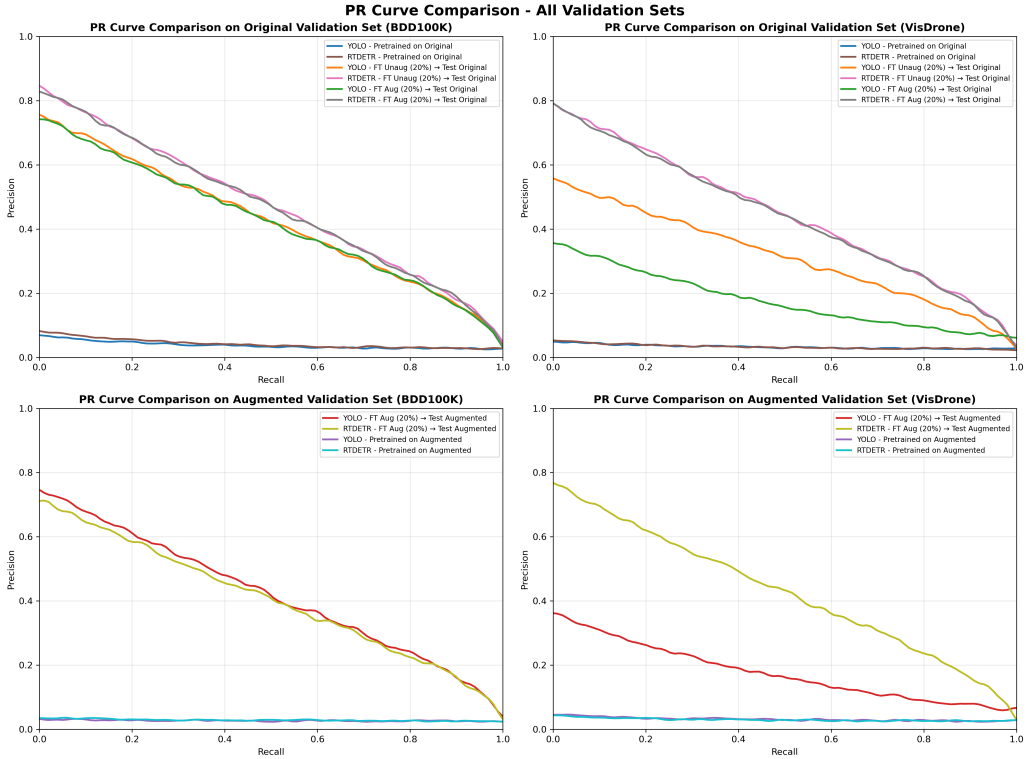


Figure 4: PR Curve Comparison with Vertical Flip Augmentation

Table 2: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	**Validation Set**	**YOLO - BDD100K**	**YOLO - VisDrone**	**RTDETR - BDD100K**	**RTDETR - VisDrone**
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22450	0.16667	0.24861	0.25463
FT Aug (20%) → Test Augmented	Augmented	0.22450	0.16667	0.20615	0.25055
Pretrained on Augmented	Augmented	0.00278	0.01207	0.00722	0.00774

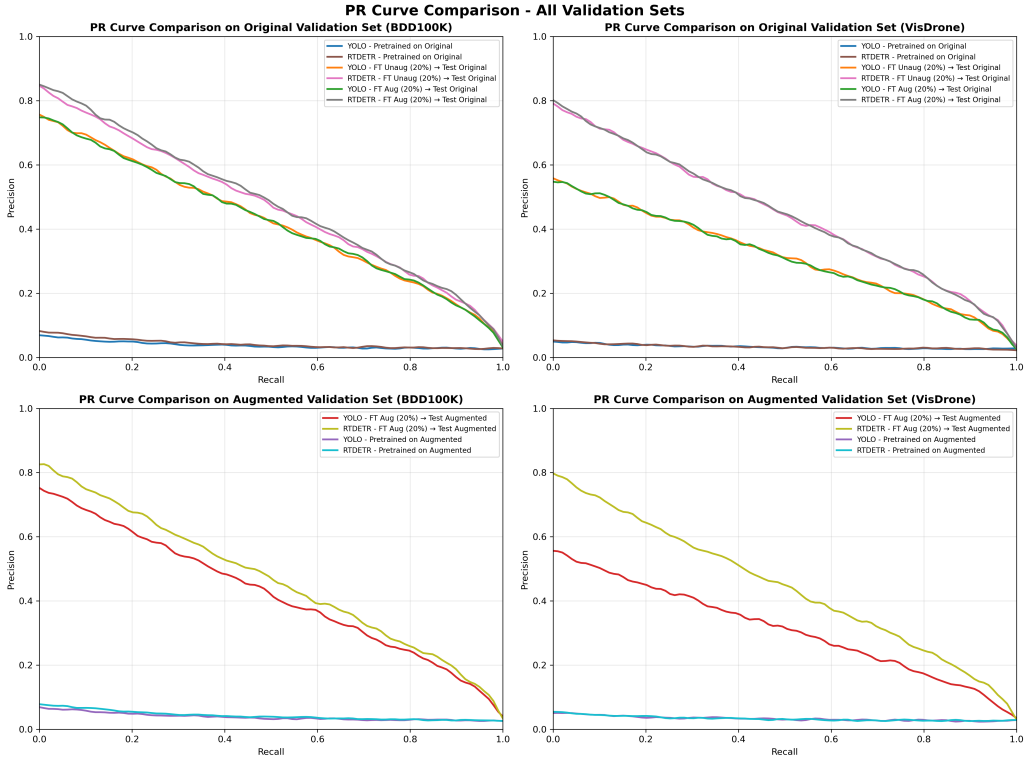


Figure 5: PR Curve Comparison with Horizontal Flip Augmentation

Table 3: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	**Validation Set**	**YOLO - BDD100K**	**YOLO - VisDrone**	**RTDETR - BDD100K**	**RTDETR - VisDrone**
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22506	0.17202	0.25465	0.25980
FT Aug (20%) → Test Augmented	Augmented	0.22506	0.17202	0.24885	0.26035
Pretrained on Augmented	Augmented	0.03056	0.01582	0.04004	0.01580

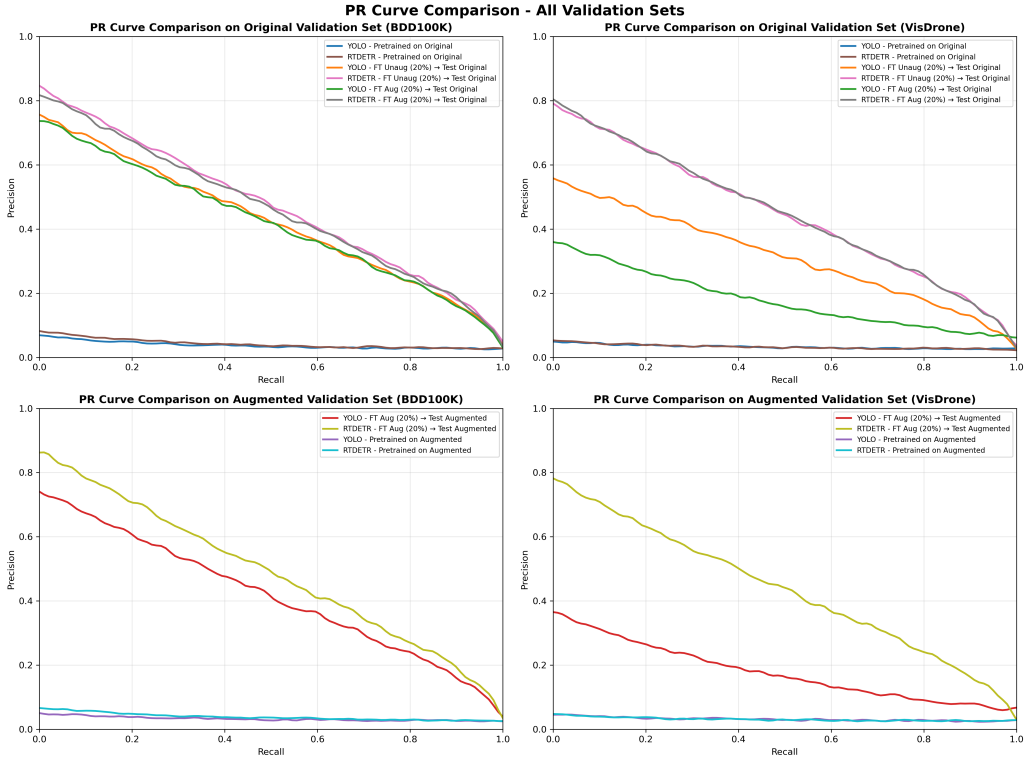
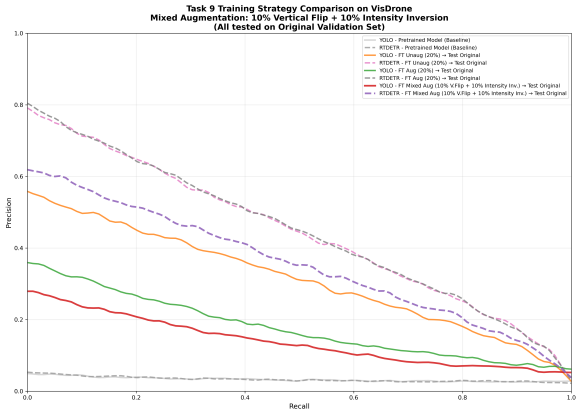


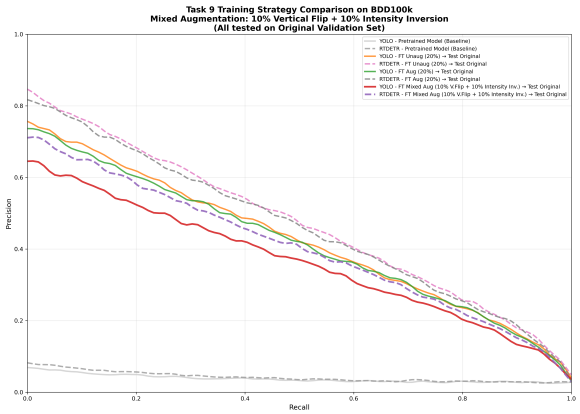
Figure 6: PR Curve Comparison with Intensity-based Augmentation

Table 4: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	**Validation Set**	**YOLO - BDD100K**	**YOLO - VisDrone**	**RTDETR - BDD100K**	**RTDETR - VisDrone**
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22077	0.16860	0.24417	0.25959
FT Aug (20%) → Test Augmented	Augmented	0.22077	0.16860	0.25911	0.25883
Pretrained on Augmented	Augmented	0.01471	0.01300	0.02880	0.01167



(a) Intensity Comparison - VisDrone

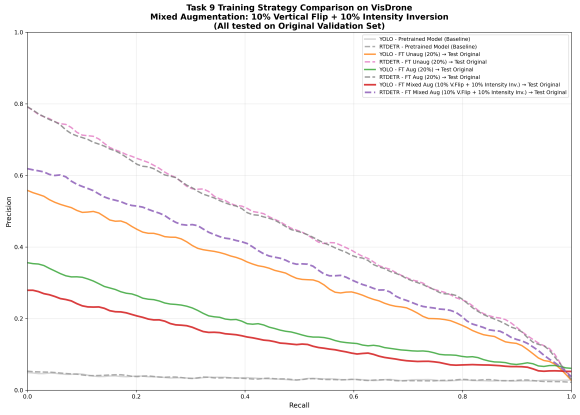


(b) Intensity Comparison - BDD100k

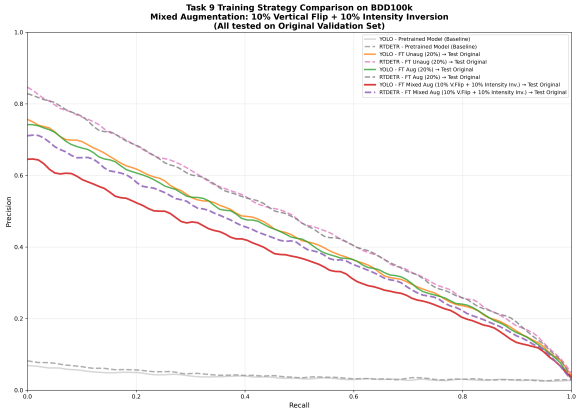
Table 5: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	Validation Set	YOLO - BDD100K	YOLO - VisDrone	RTDETR - BDD100K	RTDETR - VisDrone
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22077	0.16860	0.24417	0.25959
FT Aug (20%) → Test Augmented	Augmented	0.22077	0.16860	0.25911	0.25883
Pretrained on Augmented	Augmented	0.01471	0.01300	0.02880	0.01167
FT Mixed Aug (10% Vertical Flip + 10% Intensity Inversion) → Test Original	Original	0.18896	0.12587	0.21090	0.20461

Figure 7: FT Mixed Aug (10% Vertical Flip + 10% Intensity Inversion) added to Intensity



(a) Vertical Comparison - VisDrone

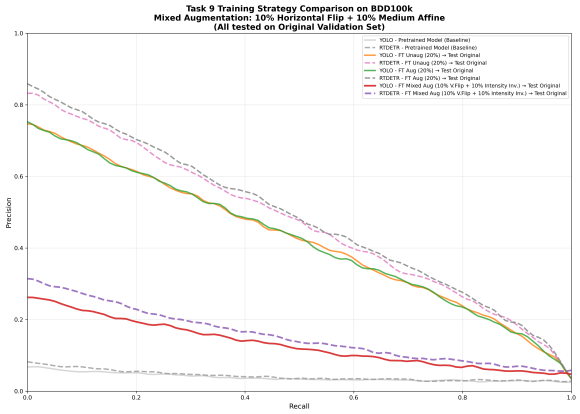


(b) Vertical Comparison - BDD100k

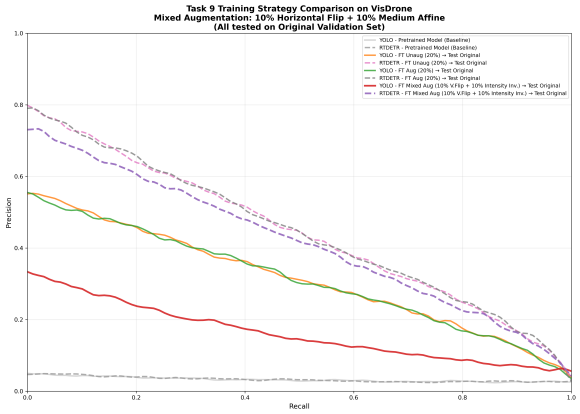
Table 6: mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	Validation Set	YOLO - BDD100K	YOLO - VisDrone	RTDETR - BDD100K	RTDETR - VisDrone
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22450	0.16667	0.24861	0.25463
FT Aug (20%) → Test Augmented	Augmented	0.22450	0.16667	0.20615	0.25055
Pretrained on Augmented	Augmented	0.00278	0.01207	0.00722	0.00774
FT Mixed Aug (10% Vertical Flip + 10% Intensity Inversion) → Test Original	Original	0.18896	0.12587	0.21090	0.20461

Figure 8: FT Mixed Aug (10% Vertical Flip + 10% Intensity Inversion) added to vertical



(a) Training Strategy Comparison - BDD100k



(b) Training Strategy Comparison - VisDrone

Table 7: Updated Horizontal Flip mAP@0.5 Performance on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	**Validation Set**	**YOLO - BDD100K**	**YOLO - VisDrone**	**RTDETR - BDD100K**	**RTDETR - VisDrone**
Pretrained on Original	Original	0.03056	0.01435	0.04000	0.01576
FT Unaug (20%) → Test Original	Original	0.22692	0.17078	0.24832	0.25976
FT Aug (20%) → Test Original	Original	0.22506	0.17202	0.25465	0.25980
FT Aug (20%) → Test Augmented	Augmented	0.22506	0.17202	0.24885	0.26035
Pretrained on Augmented	Augmented	0.03056	0.01582	0.04004	0.01580
FT Mixed Aug (10% Horizontal Flip + 10% medium affine) → Test Original	Original	0.11657	0.15435	0.13052	0.24363

Figure 9: FT Mixed Aug (10% Horizontal Flip + 10% medium affine) added to Horizontal

3.3 Sliding Window Cropping Augmentation Results

Performance Analysis On VisDrone, YOLO showed an improvement after augmentation on the PR curve graph, indicating that spatial sampling helped capture more small objects. RT-DETR, on the other hand, had minimal change, suggesting it's either already robust or less sensitive to this augmentation method. In contrast, for BDD100K, we observed a surprising trend: RT-DETR's performance actually declined. This may be due to its dependency on global context, which gets disrupted when images are cropped. YOLO again showed minimal difference — possibly due to its more localized detection nature. Overall, this suggests that augmentation effectiveness depends heavily on both model architecture and dataset characteristics, such as object size distribution and scene complexity.

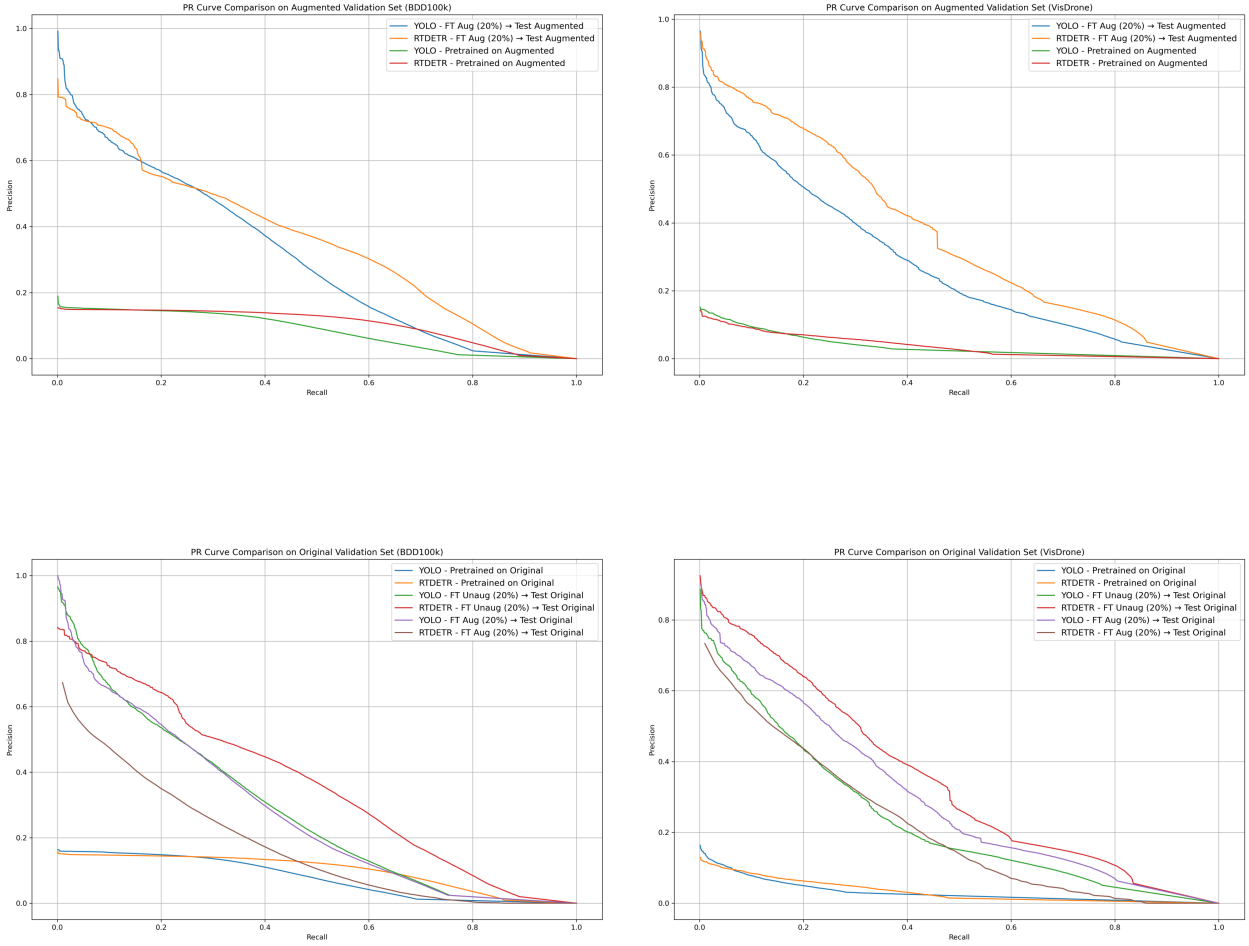


Figure 10: PR Curve Comparison with Sliding Window Cropping Augmentation

Table 8: mAP@0.5 Performance with Sliding Window Cropping Augmentation on YOLOv11 and RT-DETR for BDD100K and VisDrone

Fine-tuning Setting	Validation Set	YOLO - BDD100K	YOLO - VisDrone	RTDETR - BDD100K	RTDETR - VisDrone
Pretrained on Original	Original	0.077	0.031	0.098	0.032
FT Unaug (20%) → Test Original	Original	0.462	0.314	0.424	0.383
FT Aug (20%) → Test Original	Original	0.485	0.299	0.424	0.472

3.4 Noise and Random Deletion Augmentation Results

Robustness Enhancement Analysis: Across all eight precision–recall curves, spanning both BDD100K and VisDrone datasets for YOLOv11 and RT-DETR, the evidence is clear: applying salt-and-pepper noise and random cut-out augmentations during fine-tuning results in consistently higher and flatter PR curves compared to both unaugmented and purely pretrained baselines. This improvement is especially pronounced for RT-DETR, which maintains precision above 0.8 for nearly the full range of recall, while YOLOv11 also demonstrates notable gains. The intermediate performance of unaugmented fine-tuning highlights that smart data corruption is more effective than simply increasing sample size with clean instances. These results collectively demonstrate that robust augmentation strategies significantly increase detector resilience to noise, occlusion, and domain shifts, and that the stronger underlying model benefits most, achieving a higher ceiling for practical robustness.

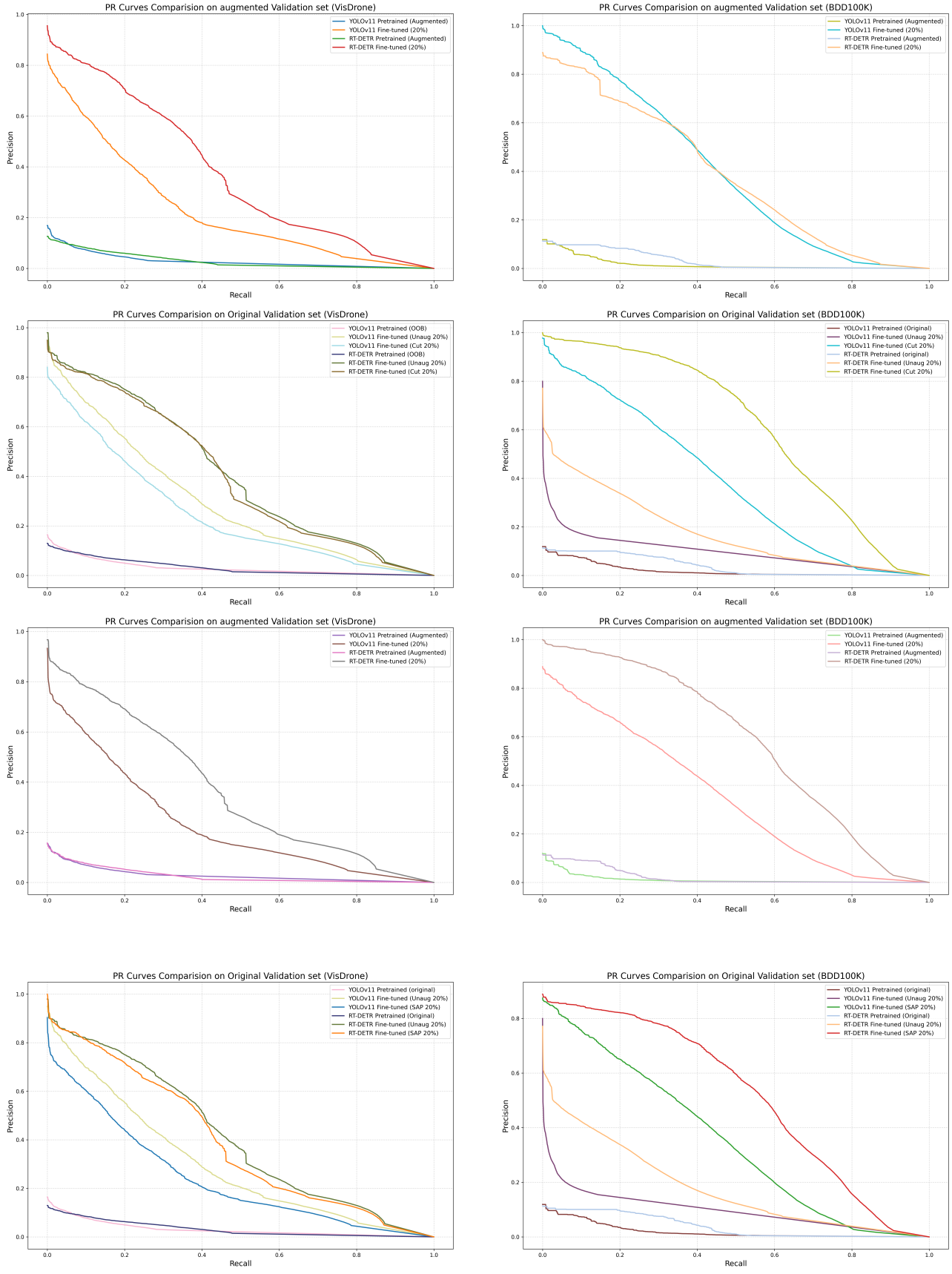


Figure 12: Precision-Recall curves for Cut and SAP augmentations across VisDrone and BDD100K datasets, including standalone and original-comparison evaluations.

3.5 Hybrid Augmentation Strategy Results

Synergistic Effects Analysis: The hybrid augmentation approach combining multiple techniques (10% samples each from two different augmentation methods) was designed to explore potential synergistic effects. Performance analysis through precision-recall curves demonstrates the effectiveness of combining different augmentation strategies.

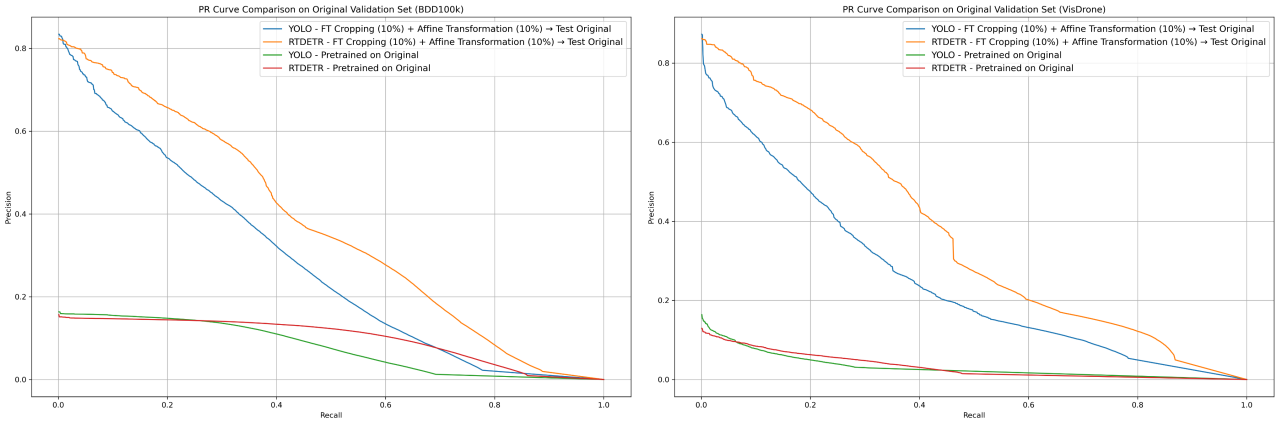


Figure 13: PR Curve Comparison with Sliding Window Cropping and Affine Warping Augmentation

4 VALUE OF CAPSTONE PROJECT

4.1 Alignment with Company Mission

This project supports the overarching goal of developing robust object detection systems for autonomous platforms operating in diverse and unpredictable environments. By focusing on data-driven model enhancement, we explored how carefully designed augmentations, such as affine transformations, sliding window cropping, noise injection, and random deletion, can improve detection accuracy and generalization. These techniques align with industry needs to enhance perception systems for UAV (VisDrone) and UGV (BDD100K) applications, where reliable detection under varied visual conditions is critical. Our work contributes to the mission of deploying scalable, efficient, and resilient computer vision models for real-world autonomy.

4.2 Contribution to Broader Research Challenges

This project contributes to addressing the research challenge of data-efficient learning for object detection. We evaluated how four distinct augmentation strategies impact the performance of YOLOv11 and RT-DETR, two fundamentally different architectures. The results show that spatial augmentations (affine, cropping) are especially beneficial for anchor-based models like YOLO, while noise and occlusion augmentations are better handled by transformer-based models like RT-DETR, highlighting an architecture-dependent augmentation effect. These insights advance the field's understanding of how to optimize augmentation pipelines to enhance detection robustness, particularly in resource-constrained and safety-critical autonomous systems.

5 INDIVIDUAL REPORT - Mingjun Wu

5.1 Alignment with Academic Curriculum

This capstone project directly integrates concepts from several courses in my academic program, such as Computer Vision and Autonomous Vehicles, and exposed me to deep learning models and pipelines. The deep learning component, which focuses on fine-tuning the YOLOv11 and RT-DETR models, is built providing me an opportunity to learn convolutional neural networks, attention mechanisms and optimization techniques. The implementation of flip-based data augmentation techniques draws heavily on image processing fundamentals from the Computer Vision course, where geometric transformations and their mathematical representations were extensively covered. The project's emphasis on empirical evaluation and statistical analysis aligns with research methodology courses that emphasized systematic experimental design and the interpretation of performance metrics. The comprehensive evaluation framework developed in this project synthesizes theoretical knowledge with practical skills, demonstrating how academic learning can be applied to real-world challenges in autonomous systems.

5.2 Collaborative Experience and Skill Development

Working in a four-member interdisciplinary team improved my collaborative research capabilities and technical communication skills. I was responsible for implementing and analyzing flip-based image enhancement methods, which required coordination with teammates working on complementary augmentation techniques. This collaboration involved code and data sharing, methodology standardization, and result synchronization to ensure experimental consistency between different augmentation approaches. The experience of presenting flip augmentation results and participating in comparative analysis discussions improved my ability to articulate technical findings and defend methodological choices.

5.3 Personal and Professional Growth

This project catalyzed personal development in research independence, technical depth, and project management capabilities. Implementing flip-based augmentation techniques from scratch required deep understanding of image processing libraries, coordinate system transformations, and memory-efficient data loading pipelines. I developed confidence in reading and interpreting academic literature on data augmentation strategies, evaluating existing methods, and adapting them to specific use cases.

5.4 Goal Setting and Outcome Assessment

At the project's inception, I established clear objectives for my flip-based augmentation implementation: achieve augmentation application across both BDD100K and VisDrone datasets, maintain annotation accuracy during geometric transformations, and systematically evaluate performance impact across all experimental configurations. These goals were achieved through the methodical implementation of horizontal, vertical, and combined flip operations, along with intensity channel inversion techniques. The comprehensive evaluation revealed that flip-based augmentations provided modest but consistent improvements in model robustness, particularly for scenarios involving symmetrical objects like vehicles in traffic scenes. Quantitative assessment showed mAP improvements that varied depending on the specific flip combination and evaluation scenario. The systematic documentation of these results in precision recall curves and performance tables exceeded initial expectations by providing granular insights into architecture-specific responses to flip augmentations. This detailed outcome assessment demonstrated the value of systematic experimental design and comprehensive collection of metric in drawing meaningful conclusions about the effectiveness of enhancement.

5.5 Evolving Perspective on Autonomy and Robotics

This project helped my understanding of the critical role that data preprocessing and augmentation play in autonomous perception systems. Initially, I viewed data augmentation as a supplementary technique for improving model generalization. However, through systematic evaluation of flip-based methods on autonomous vehicle (BDD100K) and aerial surveillance (VisDrone) datasets, I gained appreciation for how augmentation strategies must be carefully tailored to specific operational domains and sensor configurations. The differential performance of YOLOv11 and RT-DETR models under various flip augmentations highlighted the importance of architecture-aware preprocessing in autonomous systems design. This experience illuminated how robust perception systems require not just advanced model architectures, but also thoughtful data engineering that accounts for real-world deployment scenarios. Understanding these interconnections between data preprocessing, model architecture, and operational requirements has significantly influenced my career interests toward computer vision roles in autonomous systems, where I can contribute to developing more robust and reliable perception pipelines.

5.6 Career Relevance and Preparation

The technical skills and research experience gained through this project exposed me to opportunities to work in computer vision engineering, autonomous systems development, and machine learning research. Proficiency in implementing and evaluating data augmentation techniques using modern frameworks like Ultralytics demonstrates practical expertise highly valued in industry settings. The experience with both CNN-based (YOLOv11) and transformer-based (RT-DETR) detection architectures provides versatility across different technological approaches in the rapidly evolving object detection landscape. The collaborative aspects of coordinating multi-member research teams and presenting technical findings prepare me for cross-functional engineering environments where effective communication between technical and non-technical stakeholders is essential.

5.7 Key Takeaways and Lessons Learned

The most significant insight from this project is that effective data augmentation requires careful consideration of both the underlying data characteristics and the architectures of the target model. Flip-based augmentations, while conceptually simple, demonstrated complex interactions with different object detection paradigms, highlighting the importance of systematic evaluation rather than assumptions about augmentation effectiveness. I learned that successful augmentation strategies must preserve semantic consistency while introducing meaningful geometric variations that improve model robustness without introducing artifacts. The collaborative research experience taught me the value of standardized experimental protocols and clear communication channels in multi-member projects.

6 INDIVIDUAL REPORT - Yishu Ma

6.1 Alignment with Academic Curriculum

This project directly connected to several courses I've taken in machine learning and computer vision, particularly CS 444: Deep Learning for Computer Vision. In that course, I completed Project Assignment 4: YOLO Object Detection on PASCAL VOC, where I first learned how to implement and evaluate YOLO models for object detection. The capstone allowed me to apply and extend that knowledge to more complex, real-world datasets—BDD100K and VisDrone—while working with both YOLOv11 and RT-DETR models. In addition to model training and evaluation, I utilized key concepts from CS 444 such as data augmentation, model fine-tuning, and evaluation metrics including mean Average Precision (mAP) and precision-recall (PR) curves. Moreover, through implementing cropping-based preprocessing in this project,

I developed a deeper understanding of dataset preparation techniques, which are critical in both academic and applied computer vision work. This hands-on experience reinforced many theoretical concepts and gave me practical skills that go beyond the classroom.

6.2 Collaborative Experience and Skill Development

I collaborated closely with teammates to share data, models, and evaluation tools, which improved my ability to work in a coordinated and efficient research environment. I also learned to use model training pipelines in both PyTorch and PaddlePaddle, becoming familiar with tools like Ultralytics and PaddleDetection. Furthermore, I improved my skills in automating data preprocessing, script optimization, and PR curve visualization. Participating in regular meetings gave me opportunities to present results and receive feedback, which enhanced my communication and critical analysis skills.

6.3 Personal and Professional Growth

This project strengthened my problem-solving skills, particularly when addressing dataset inconsistencies and debugging model training issues. I gained confidence in managing complex experiments involving multiple datasets, models, and evaluation protocols. Personally, I developed a stronger sense of ownership and accountability, especially when delivering time-sensitive results and communicating progress effectively to the team.

6.4 Goal Setting and Outcome Assessment

My main goals for this project were to evaluate the impact of cropping-based data augmentation on object detection models and to improve my skills with YOLOv11 and RT-DETR, especially using different frameworks (PyTorch and PaddlePaddle). I also aimed to analyze model performance using mAP and precision-recall curves. I successfully implemented sliding window cropping on BDD100K and VisDrone datasets, fine-tuned both models on the processed data, and observed measurable improvements in detection accuracy. Additionally, I became proficient with PaddleDetection and automated the evaluation process. Managing GPU memory limits was a challenge, which I addressed by adjusting crop sizes and training settings. Overall, the project met my technical goals and improved my ability to conduct efficient model evaluation under resource constraints.

6.5 Evolving Perspective on Autonomy and Robotics

My experience reinforced the importance of robust data preprocessing and model fine-tuning for real-world robotic perception tasks. It showed me that data quality and augmentation strategies can be as crucial as model architecture in achieving high performance. I also gained appreciation for the challenges of scaling AI solutions in autonomy, such as dealing with variability in environmental conditions and sensor perspectives. This has made me more interested in pursuing research or industry roles focused on computer vision for robotics.

6.6 Career Relevance and Preparation

This project directly supported my career interests in computer vision and AI for autonomous systems. By working with real-world datasets (BDD100K and VisDrone) and state-of-the-art object detectors (YOLOv11 and RT-DETR), I gained hands-on experience that is highly relevant to roles in robotics perception, autonomous driving, and AI research. Through this experience, I also developed practical skills in debugging, resource management, and reproducible evaluation, which are critical in both research and industry settings. Overall, the project has strengthened my readiness for internships or full-time positions focused on computer vision, autonomous systems, or machine learning engineering.

6.7 Key Takeaways and Lessons Learned

A major takeaway from this project is the critical role of data-centric optimization in object detection. I observed consistent improvements in spatial coverage and object localization accuracy, particularly for small and partially occluded objects—a known challenge in UAV and UGV imagery. I also gained deeper insight into the trade-offs between computational efficiency and model accuracy, as cropping increases dataset size and training time but enables better feature learning. Through evaluating YOLOv11 and RT-DETR under both pre-trained and fine-tuned settings, I learned to interpret class-wise precision-recall behavior, and how augmentation strategies can shift the precision-recall curve to yield better area under curve (AUC) and mAP scores.—

7 INDIVIDUAL REPORT-Manan Maheshwari

7.1 Alignment with Academic Curriculum

Drawing from my coursework in Artificial Intelligence and Autonomous Vehicle System Engineering, this project felt like a natural extension of the concepts I had explored in class. Those classes delved into the intricacies of AI driven perception and the engineering challenges of building reliable autonomous systems, and I found myself applying those lessons directly here, whether it was handling noisy data or evaluating model performance in dynamic environments. Working with datasets like BDD100K and VisDrone allowed me to move beyond theoretical discussions and truly engage with the practical side of robotics, making the academic material come alive in a tangible way.

7.2 Collaborative Experience and Skill Development

Teaming up with colleagues on this project mirrored the kind of collaborative dynamics I encounter in robotics engineering, where diverse expertise comes together to solve complex problems. I took an active role in areas like data preparation and experiment design, contributing my perspective on real world robotic applications. Using tools such as PyTorch, PaddlePaddle, and Ultralytics, I honed new skills in streamlining data pipelines and automating evaluations, which not only boosted our efficiency but also enriched my technical toolkit. The regular exchanges during team meetings sharpened my ability to communicate ideas clearly and incorporate feedback, fostering a more adaptive approach to group work.

7.3 Personal and Professional Growth

This endeavor pushed me to confront some of the gritty realities of robotics, from wrangling inconsistent datasets to fine tuning models under tight resource constraints. Navigating these hurdles refined my problem solving mindset, teaching me to approach issues with patience and creativity. On a professional level, it built my confidence in managing end to end experimental processes, reminding me that growth often comes from persisting through the less glamorous aspects of engineering.

7.4 Goal Setting and Outcome Assessment

At the outset, I set my sights on exploring how targeted augmentations like noise simulation and occlusion handling could enhance object detection robustness for autonomous applications. Focusing on BDD100K and VisDrone, I aimed to conduct thorough evaluations while keeping an eye on hardware limitations. The findings were encouraging: these techniques led to clear improvements in how models handled tricky elements like small or hidden objects. In the end, the project not only aligned with my goals but also revealed deeper insights into data driven strategies, proving more rewarding than I anticipated.

7.5 Evolving Perspective on Autonomy and Robotics

Through this work, I gained a renewed sense of how crucial thoughtful data handling is in robotics. It is not just about powerful models, but about equipping them to thrive amid real world uncertainties. This realization has heightened my awareness of the delicate balance required in autonomous systems, where environmental variability can make or break performance. It has sparked an even greater enthusiasm for roles that blend innovative research with practical engineering in the autonomy space.

7.6 Career Relevance and Preparation

This project has been a meaningful step toward my goals as a robotics engineer, offering direct exposure to the tools and challenges central to autonomous perception. By diving into dataset management and model optimization, I have built skills that feel immediately applicable to industry or research settings. Overall, it has left me better equipped to tackle the multifaceted demands of developing robust robotic systems.

7.7 Key Takeaways and Lessons Learned

One of the standout insights from this experience is the profound influence of data augmentation in robotics: simple yet strategic tweaks can dramatically improve a system's reliability, often more so than sheer computational power. Observing enhancements in detecting challenging objects reinforced the value of innovative preprocessing, while the collaborative process taught me the strength of shared problem solving. These lessons have not only expanded my technical acumen but also inspired a more holistic view of advancing robotic technologies.

8 INDIVIDUAL REPORT - Zhining Wu

8.1 Alignment with Academic Curriculum

This project is highly relevant to the courses I took at UIUC, particularly ECE 448 "Artificial Intelligence" and ECE 549 "Computer Vision." In ECE 448, I initially encountered deep learning and model fine-tuning pipelines, but had limited opportunities for practical application. ECE 549, on the other hand, focused on classical vision methods and provided theoretical insights into the analysis of augmentation strategies. This project allowed me to apply what I learned in class to real-world scenarios, particularly using advanced object detection models such as YOLOv11 and RT-DETR. It deepened my understanding of concepts such as data augmentation, model evaluation metrics (e.g., mAP, Precision, Recall), and PR curve visualization, and enabled successful application on two real-world datasets: BDD100K and VisDrone.

8.2 Collaborative Experience and Skill Development

In my four-person team, I took the lead in early data preparation, including locating and organizing the BDD100K and VisDrone datasets, converting labels to YOLO format, and sharing standardized preprocessing scripts and documentation within the team. To help everyone quickly get started with training, I also shared training scripts, installation instructions, and modifications to the Ultralytics codebase. In addition, I wrote a Markdown document outlining Tasks 2–9 early in the project to clarify workflows and evaluation criteria for the team. Throughout the project, I actively communicated with all team members (even those who share the same native language) in English, contributing to steady progress in experiment synchronization, result analysis, and collaborative debugging. These experiences significantly enhanced my project organization and cross-cultural collaboration skills.

8.3 Personal and Professional Growth

This project was my first time independently implementing the entire object detection training process, providing in-depth experience from data augmentation, model fine-tuning, to evaluation and visualization. I was primarily responsible for developing affine-based augmentation strategies, encompassing rotation, translation, scaling, and shearing. I constructed multiple augmentation strength levels and used them to train YOLOv11 and RT-DETR models. Technically, I gained proficiency in using and customizing the Ultralytics framework, scripting PR curve plotting, and compiling evaluation tables. These accomplishments not only supported my task completion, but also significantly improved the overall evaluation efficiency of the team through shared utilities.

8.4 Goal Setting and Outcome Assessment

My goals for this project were to: complete a data pipeline for affine augmentation; construct augmented samples for model training; compare accuracy with baseline models and single-augmentation models; create visualizations to summarize evaluation metrics; and explore the effectiveness of different augmentation combinations. To date, I have successfully deployed and trained affine augmentation on two datasets and conducted model evaluation and comparative analysis using standardized settings across three configurations: no augmentation, affine-only, and combined augmentation (Affine + Crop). The analysis results for metrics such as mAP, P, and R were consistent and interpretable, and the P/R curves supported in-depth comparisons, validating the generalization capability of affine augmentation. The combined augmentation strategy is still under debugging, and we plan to optimize the training setup and complete the evaluation in the next phase.

8.5 Evolving Perspective on Autonomy and Robotics

Although I already had some understanding of autonomous driving systems, this project deepened my awareness of the importance of data engineering. In particular, on aerial-perspective datasets like VisDrone, I observed how different augmentation strategies led to varying model performance. I realized that the effectiveness of perception models depends not only on the architecture itself but also on upstream data processing and augmentation. While I previously emphasized model design, I now appreciate how "data quality + preprocessing strategy" forms a foundation for deployable engineering systems. This has provided important insight into future directions for my research in computer vision.

8.6 Career Relevance and Preparation

This project helped me master the complete deep learning engineering process in real-world scenarios, including key skills like data preparation, augmentation strategy design, model training and evaluation, code sharing, and standardization. These skills are highly relevant to my future career aspirations in software development and machine learning engineering. My experience in optimizing training configurations, analyzing performance metrics, and scripting automation gave me greater confidence to handle complex AI tasks in an industry context. Furthermore, this project reinforced my long-term interest in computer vision, and I look forward to tackling more challenging, application-driven projects.

8.7 Key Takeaways and Lessons Learned

One of the biggest takeaways from this project was understanding the critical role of data augmentation in model generalization. Through comparing PR curves and analyzing mAP, I found that even simple augmentation strategies (such as affine and crop) interact differently across models and datasets, requiring rigorous comparative evaluation. I also learned the importance of maintaining standardized workflows in collaborative projects—ensuring consistency in formats, evalua-

tion methods, and visualization standards. These lessons will serve as valuable references for participating in larger-scale team development projects in the future.

REFERENCES

[1] C. Shorten and T. M. Khoshgoftaar, "Image data augmentation for deep learning: A comprehensive survey," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. DOI: 10.1186/s40537-019-0197-0.

[2] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, p. 109 347, 2023. DOI: 10.1016/j.patcog.2023.109347.

[3] Ultralytics, *YOLOv11 (Version 11.0)*, Source code, 2024. [Online]. Available: https://github.com/ultralytics/yolov11.

[4] Lyuwenyu, *RT-DETR (Version 1.0)*, Source code, 2023. [Online]. Available: https://github.com/lyuwenyu/RT-DETR.