Enhancing Real-World Object Detection with YOLOv8 and MLOps: Applications in Urban Safety

Jordy Romero Armijos and Nayeli Barbecho Cajamarca

Universidad Politécnica Salesiana, Ecuador

Abstract. This paper presents a comprehensive architecture for object detection focused on three critical classes for daily life safety: *person*, *car*, and *dog*. These classes represent the most dynamic agents in urban environments, making their accurate detection vital for autonomous driving and smart surveillance. The system integrates the YOLOv8 model with advanced data augmentation techniques, specifically Copy-Paste and MixUp, to address intra-class variability. Furthermore, a robust MLOps strategy is implemented using MLflow for experiment tracking and model registry management. The results demonstrate that hyperparameter optimization allows the global mAP50 to exceed 0.70. Notably, detection performance for the *dog* class improved significantly, validating the system's applicability in real-world, safety-critical environments.

Keywords: YOLOv8 · Urban Safety · MLOps · Object Detection · Autonomous Systems · Pascal VOC

1 Introduction

Object detection has become a fundamental component of modern computer vision systems, particularly in safety-critical domains such as autonomous driving, smart cities, and intelligent surveillance. Despite substantial advances in deep learning, building models capable of generalizing effectively in uncontrolled urban environments remains a complex challenge due to occlusions, lighting variations, scale changes, and unpredictable object motion [4].

Urban environments are dominated by highly dynamic agents such as pedestrians, vehicles, and domestic animals. While person and car detection are widely studied tasks, the inclusion of animals such as dogs introduces additional complexity. Dogs present high intra-class variability, irregular and non-linear motion, and deformable body structures, making them especially difficult to detect under real-world conditions. Their presence in traffic environments can also lead to high-risk scenarios.

Recent developments in real-time object detection architectures, particularly within the YOLO (You Only Look Once) family, have significantly improved the balance between computational efficiency and detection accuracy. YOLOv8, introduced by Ultralytics [2], incorporates architectural refinements, anchor-free detection strategies, and optimized training pipelines that enhance performance across multiple benchmarks.

This work proposes a production-ready object detection pipeline based on YOLOv8, optimized for three urban-relevant classes: *person*, *car*, and *dog*. Beyond model training, a structured MLOps workflow using MLflow is integrated to ensure reproducibility, experiment traceability, and controlled model management [5,3]. The primary objective is to design a robust and deployable detection system tailored for real-world urban safety applications.

2 Proposed Method

The proposed methodology follows a structured production-oriented workflow that integrates data preprocessing, model training, optimization, and experiment management.

2.1 System Architecture

The architecture is composed of three main interconnected modules:

1. **Data Module:** Responsible for dataset validation, preprocessing, and annotation conversion. Pascal VOC XML annotations are converted into YOLO-compatible format to enable efficient training [1].
2. **Training Module:** Implements transfer learning using YOLOv8 with pre-trained COCO weights. Training is performed using PyTorch, incorporating hyperparameter tuning to improve convergence stability and detection accuracy [2].
3. **Management Module (MLOps):** MLflow is employed for experiment tracking, parameter logging, metric monitoring, and model registry control. This ensures reproducibility and systematic evaluation of model improvements [5].

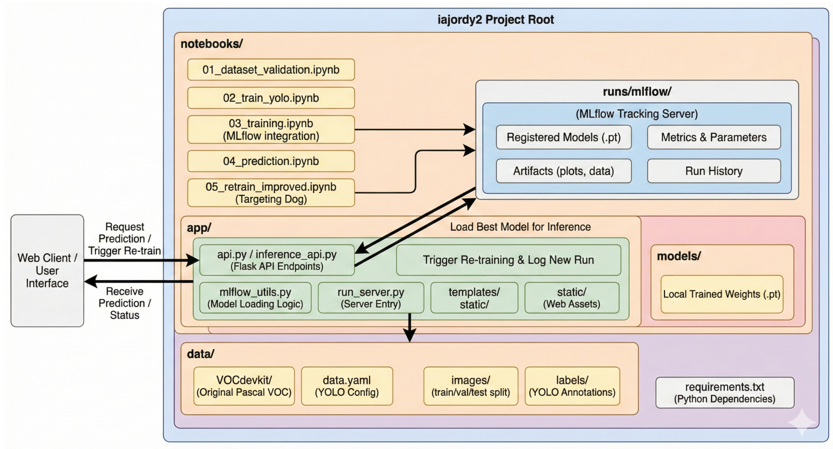

Fig. 1. System architecture integrating data preprocessing, YOLOv8 training, and MLflow-based experiment management

2.2 Algorithms and Techniques

The model was initialized using pretrained YOLOv8 weights provided by Ultralytics, originally trained on the COCO dataset. This transfer learning strategy allows leveraging generalized feature representations while fine-tuning the detector for three specific urban classes: *person*, *car*, and *dog*.

Two configurations were evaluated: a baseline model (YOLOv8n) and an improved model (YOLOv8s). The improved version incorporates extended training epochs, modified batch size, and targeted augmentation strategies to enhance performance on deformable and unpredictable objects.

The following augmentation techniques were employed:

– **Mosaic and MixUp:** Standard augmentations within the YOLO training pipeline that improve robustness against occlusions and contextual variability.
– **Copy-Paste Augmentation (0.3 ratio):** Introduced in the improved model to artificially increase the presence of dog instances in diverse backgrounds, enhancing generalization for small and deformable targets.

Additionally, the Non-Maximum Suppression (NMS) IoU threshold was adjusted from 0.70 to 0.50 in the improved configuration to reduce missed detections in crowded scenes.

3 Design of Experiments

Experiments were conducted using the Pascal VOC 2012 dataset [1], focusing exclusively on three selected classes. The dataset was divided into training and validation subsets following standard evaluation protocols.

Model performance was evaluated using mean Average Precision at IoU threshold 0.50 (mAP50) and the stricter mAP50–95 metric, which averages performance across multiple IoU thresholds.

Table 1. Urban relevance and detection complexity of selected classes

Class	Urban Relevance	Detection Complexity
Person	Pedestrian safety	Medium
Car	Traffic regulation	Low
Dog	Accident prevention	High

Hyperparameter optimization was conducted through iterative experimentation tracked in MLflow, enabling systematic comparison between baseline and improved configurations.

4 Results and Discussion

The optimized configuration demonstrated significant performance improvements compared to the baseline model. The global mAP50 increased from 0.602 to 0.715, while mAP50–95 improved from 0.354 to 0.482.

Table 2. Baseline vs. optimized model performance

Model	mAP50	mAP50–95
Baseline	0.602	0.354
Optimized	0.715	0.482

The most notable improvement was observed in the *dog* class, where AP50 increased from 0.46 to 0.65. This confirms that targeted augmentation strategies such as Copy-Paste are particularly effective for small, deformable, and unpredictable objects.

From a practical perspective, improving detection reliability for animals in urban traffic scenarios can significantly reduce accident risk in autonomous driving systems.

Furthermore, the integration of MLflow provided structured experimentation, enabling transparent comparison of model variants and ensuring reproducibility—an essential requirement for production-level AI systems.

5 Conclusions

This study demonstrates that combining YOLOv8 with targeted data augmentation strategies and structured MLOps practices results in substantial improvements in object detection performance for urban safety applications.

The most challenging class, dogs, benefited significantly from Copy-Paste augmentation, validating its effectiveness for deformable and underrepresented objects. The improvement in global mAP metrics confirms that systematic hyperparameter tuning and experiment tracking contribute meaningfully to performance gains.

Beyond detection accuracy, the incorporation of MLflow ensures reproducibility, traceability, and scalability, making the proposed pipeline suitable for real-world deployment in intelligent transportation and surveillance systems.

Future work may include real-time deployment testing, cross-dataset generalization evaluation, and the extension of the system to additional urban-critical object classes.

References

1. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010). `https://doi.org/10.1007/s11263-009-0275-4`
2. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics yolov8. `https://github.com/ultralytics/ultralytics` (2023), gitHub repository
3. Kreuzberger, D., Kühl, N., Hirschl, S.: Machine learning operations (mlops): Overview, definition, and architecture. IEEE Access **11**, 31866–31879 (2023). `https://doi.org/10.1109/ACCESS.2023.3262418`
4. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 60 (2019). `https://doi.org/10.1186/s40537-019-0197-0`
5. Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S.A., Konwinski, A., et al.: Accelerating the machine learning lifecycle with mlflow. IEEE Data Engineering Bulletin **41**(4), 39–45 (2018)