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

Jordy Romero Armijos and Nayeli Barbecho Cajamarca

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 metric tracking and model registry. The results demonstrate that hyperparameter optimization allows the global mAP50 to exceed 0.70. Crucially, the detection of the "Dog" class—often a cause of unpredictable accidents—achieved significant improvement, validating the system's potential for real-world deployment in dynamic environments.

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

1 Introduction

Computer vision plays a fundamental role in modern process automation. However, developing robust models faces the challenge of generalizing in uncontrolled environments. In the context of daily life utility, the interaction between pedestrians, vehicles, and animals defines the complexity of modern cities.

Detecting *Persons* and *Cars* is standard for traffic monitoring, but the inclusion of *Dogs* introduces a critical variable for autonomous driving systems and smart home security. An autonomous vehicle must brake for a pet just as it does for a human, yet animals are often smaller and move erratically. Similarly, home security cameras must distinguish a resident's pet from an intruder or a vehicle to prevent false alarms.

This work addresses the implementation of a detection system based on YOLOv8 (You Only Look Once version 8) [2,6], selected for its real-time inference capabilities suitable for edge devices. The project establishes a complete *pipeline* integrating MLflow [3] to optimize the detection of these specific classes, ensuring that the model is not only accurate but reliable for practical implementation in safety-critical scenarios [4, 10].

2 Proposed Method

The methodology is structured in a linear workflow spanning from data ingestion to model deployment, designed to emulate a production environment.

2.1 System Architecture

The system is composed of three interconnected modules:

1. **Data Module:** Responsible for validating the dataset. XML annotations are converted to YOLO format [1].
2. **Training Module (Core):** Utilizes PyTorch and Ultralytics [7]. The CNN is defined here.
3. **Management Module (MLOps):** MLflow acts as the administrative brain, logging executions and managing the model registry to ensure that only the safest model is deployed.

2.2 Algorithms and Techniques

The training implements *Transfer Learning* from COCO. To solve the real-world problem of "unpredictable targets" (like dogs), specific augmentation algorithms are introduced [5]:

- **Mosaic & Mixup:** Improve robustness against occlusions common in busy streets.
- **Copy-Paste:** Copies instances of objects (especially dogs) onto different backgrounds to simulate diverse environments [4].

3 Design of Experiments

To validate the proposal for daily utility, two scenarios are defined using the Pascal VOC 2012 dataset [1].

3.1 Dataset Context

The selected classes represent the "Urban Trinity" of obstacles:

Table 1: Instance Distribution and Real-World Relevance

Class	Relevance	Complexity
Person	Pedestrian Safety	Medium
Car	Traffic Control	Low (Rigid)
Dog	Accident Prevention	High (Deformable)

3.2 Experimental Configuration

We compare a baseline model against an improved version optimized for these specific real-world challenges (Table 2) [8].

Table 2: Hyperparameter Comparison

Parameter	Baseline (V8n)	Improved (V8s)
Structure	Nano	Small
Epochs	50	100
Batch Size	16	8
Optimization	Standard	Targeted Augmentation
Copy-Paste	0.0	0.3
NMS IoU	0.70	0.50

4 Results and Discussion

The results (Table 3) highlight the practical viability of the system.

The optimized model shows a drastic improvement. In a real-world context, the increase in **Dog AP50 from 0.46 to 0.65** implies a significantly higher probability of an autonomous car detecting a pet running across the street, potentially preventing accidents. The high precision in **Person (0.86)** and **Car (0.79)** confirms the model's reliability for standard surveillance tasks. The deployment via Flask [9] allows this model to be integrated into web or mobile apps for immediate use.

Table 3: Performance Results by Model

Model	mAP50	mAP50-95	P(Person)	P(Car)	P(Dog)
Baseline	0.602	0.354	0.78	0.70	0.46
Improved	**0.715**	**0.482**	**0.86**	**0.79**	**0.65**

5 Conclusions

The realization of this project consolidated key knowledge regarding computer vision applications in daily life. The most significant challenge was the optimization of the "Dog" class; it was observed that small, erratic targets are the most dangerous in automated scenarios. However, it was demonstrated that using Copy-Paste augmentation directly impacted the model's ability to generalize these targets, making the system safer for real-world deployment.

Furthermore, the integration of MLflow transformed the workflow. It was learned that for a system to be useful in production (e.g., a smart city camera), strict version control is required to prevent regression. Finally, it was concluded that YOLOv8 provides the necessary speed for real-time protection of pedestrians and pets, meeting the safety objectives established at the beginning of the research.

References

[1] M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, 2010.

[2] G. Jocher et al., "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[3] M. Zaharia et al., "Accelerating the Machine Learning Lifecycle with MLflow," *IEEE Data Eng. Bull.*, 2018.

[4] G. Ghiasi et al., "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," in *CVPR*, 2021.

[5] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, 2019.

[6] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in *CVPR*, 2016.

[7] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *NeurIPS*, 2019.

[8] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2019.

[9] M. Grinberg, *Flask Web Development.* O'Reilly Media, 2018.

[10] D. Kreuzberger et al., "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, 2023.