Indonesian Car Retrieval System: An End-to-End Deep Learning Approach for Vehicle Detection and Type Classification

Muhammad Hafiz
Universitas Syiah Kuala
Banda Aceh, Indonesia
techhafiz81@gmail.com

***Abstract*—This paper presents a comprehensive Car Retrieval System designed for Indonesian vehicle detection and classification. The system integrates YOLOv8 object detection with a ResNet50-based classifier to identify eight distinct car types commonly found in Indonesia. Our approach achieves 86.33% classification accuracy on a dataset of over 17,000 images spanning crossover, hatchback, MPV, offroad, pickup, sedan, truck, and van categories. The integrated pipeline processes video at 13.6 FPS, demonstrating real-time capability. Experimental results show strong per-class performance with F1-scores ranging from 0.83 to 0.91 across all vehicle categories. We provide a detailed analysis of the model's performance, including confusion matrix evaluation and per-class metric breakdowns, highlighting the system's robustness and identifying areas for future improvement.**

***Index Terms*—Object Detection, Car Classification, Deep Learning, YOLOv8, ResNet50, Transfer Learning, Indonesian Vehicles**

I. Introduction

The rapid expansion of road networks and the increasing volume of vehicular traffic in Indonesia necessitate advanced intelligent transportation systems (ITS). Automated vehicle identification and classification are fundamental components of modern ITS, enabling applications such as electronic toll collection, traffic flow analysis, and law enforcement automation. In the Indonesian context, understanding the distribution and types of vehicles on roads is particularly crucial for traffic management and urban planning, given the unique vehicle composition of the region.

This paper addresses the specific challenge of developing an end-to-end car retrieval system tailored for the Indonesian automotive landscape. The primary objectives of this research are:

1) To develop a robust object detection module capable of identifying multiple vehicle instances in complex visual environments, including static images and dynamic video streams.
2) To implement a high-accuracy classification system that categorizes detected vehicles into eight specific Indonesian car types: crossover, hatchback, MPV, offroad, pickup, sedan, truck, and van.
3) To integrate these components into a unified, real-time pipeline suitable for practical deployment scenarios.

We propose a two-stage deep learning pipeline combining state-of-the-art object detection architecture with a custom-trained classification network. The YOLOv8 detector [1] is employed to localize vehicle regions with high speed and precision. Subsequently, a ResNet50-based classifier [3], optimized via transfer learning, determines the specific car type for each detected instance. This modular approach allows for independent optimization of detection and classification tasks, resulting in a flexible and high-performance system.

II. Related Work

A. Object Detection

The field of object detection has witnessed a paradigm shift from traditional computer vision techniques, such as Histogram of Oriented Gradients (HOG) combined with Support Vector Machines (SVM), to deep learning-based approaches. One-stage detectors, particularly the YOLO (You Only Look Once) family [2], have revolutionized real-time detection by framing the problem as a single regression task, mapping image pixels directly to bounding box coordinates and class probabilities. The latest iteration, YOLOv8, introduces significant architectural enhancements, including the C2f module for richer gradient flow and an anchor-free detection head, which further improves accuracy and inference speed, making it an ideal candidate for real-time traffic monitoring systems.

B. Vehicle Classification

Vehicle make and model recognition (VMMR) and type classification have been extensively studied using Convolutional Neural Networks (CNNs). Deep architectures like VGG, Inception, and ResNet [3] have become the de facto standards for these tasks. ResNet, in particular, introduced residual connections that mitigate the vanishing gradient problem, enabling the training of substantially deeper networks. Transfer learning, where models pretrained on large-scale datasets like ImageNet [4] are fine-tuned on domain-specific data, has proven highly effective for vehicle classification tasks with limited labeled data. Recent works have also explored attention mechanisms and Vision Transformers (ViT) to capture fine-grained features, though CNNs remain widely used for their balance of performance and computational efficiency.

III. Methodology

A. System Architecture

Our proposed system architecture follows a sequential two-stage pipeline, maximizing the strengths of specialized models for detection and classification tasks. The workflow is illustrated in Figure 1.

1) **Detection Stage**: We utilize the YOLOv8-nano model, pretrained on the COCO dataset. This lightweight yet powerful model filters the input stream for vehicle-related classes (car, bus, truck), generating bounding boxes for all potential vehicle instances. The choice of the 'nano' variant ensures low latency, essential for video processing.
2) **Preprocessing Stage**: Detected regions of interest (ROIs) are cropped from the original frame. These crops undergo preprocessing steps, including resizing to 224×224 pixels and normalization using ImageNet mean and standard deviation values, to prepare them for the classifier.
3) **Classification Stage**: A ResNet50 model handles the type categorization. We modify the final fully connected layer to output probabilities for our eight specific target classes. The model takes the preprocessed vehicle crops and assigns a class label based on the highest confidence score.

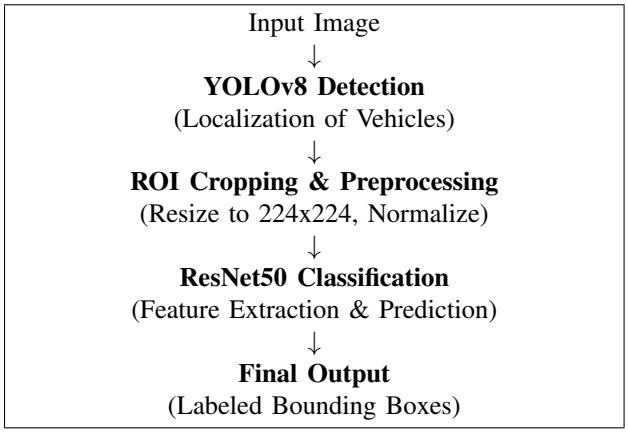

Fig. 1. Detailed System Architecture showing the data flow from input to final annotated output.

B. Dataset Description

The development and evaluation of our system rely on a comprehensive dataset of Indonesian vehicles. The dataset comprises 17,171 images distributed across eight distinct categories, representing the most common vehicle types on Indonesian roads. The distribution of the dataset is detailed in Table I.

The dataset exhibits a slight class imbalance, with MPVs being the most dominant class (3,124 images) and Trucks being the least represented (824 images). This distribution reflects the real-world prevalence of these vehicle types in Indonesia.

TABLE I
Dataset Distribution across Train, Validation, and Test Splits

Class	Train	Val	Test	Total
Crossover	2,125	265	266	2,656
Hatchback	2,090	309	280	2,679
MPV	2,469	342	313	3,124
Offroad	1,919	240	240	2,399
Pickup	744	153	123	1,020
Sedan	1,961	284	255	2,500
Truck	659	83	82	824
Van	1,575	197	197	1,969
Total	**13,542**	**1,873**	**1,756**	**17,171**

C. Classification Model Details

For the classification task, we employ a ResNet50 architecture. The network is modified to adapt to our specific 8-class problem:

- **Backbone**: The layers of ResNet50 up to the final global average pooling layer are used as a feature extractor. We initialize these layers with weights pretrained on ImageNet to leverage learned low-level features.
- **Classification Head**: We replace the original fully connected layer with a custom head consisting of:
  - A Flatten layer.
  - A Dropout layer with $p = 0.5$ to prevent overfitting.
  - A Linear layer mapping 2048 input features to 512 hidden units, followed by ReLU activation.
  - A second Dropout layer ($p = 0.25$).
  - A final Linear layer mapping 512 units to the 8 output classes.

D. Training Strategy

The model was trained using a rigorous setup to ensure optimal convergence and generalization. We employed the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, starting with a learning rate of 10^{-4}. A Cosine Annealing learning rate scheduler was implemented to gradually reduce the learning rate to 10^{-6} over 20 epochs. To further improve robustness, we utilized Label Smoothing with a factor of 0.1, preventing the model from becoming overconfident in its predictions.

Data Augmentation: To address potential overfitting and improve the model's ability to handle diverse real-world conditions, we applied extensive data augmentation during training:

- Random Resized Crop: To enforce scale invariance.
- Random Horizontal Flip: To handle orientation variations.
- Color Jitter: Random adjustments to brightness, contrast, saturation, and hue to simulate different lighting conditions.
- Random Rotation: Rotations up to $\pm 15^\circ$ to handle slight view angle changes.
- Random Erasing: To simulate occlusions.

IV. Experimental Results

A. Classification Performance

The trained classifier was evaluated on the held-out test set containing 1,756 images. Table II summarizes the classifica-

tion metrics for each class.

TABLE II
DETAILED CLASSIFICATION METRICS ON TEST SET

Class	Precision	Recall	F1-Score
Crossover	0.85	0.81	0.83
Hatchback	0.88	0.83	0.85
MPV	0.80	0.90	0.85
Offroad	0.86	0.85	0.86
Pickup	0.93	0.89	0.91
Sedan	0.87	0.87	0.87
Truck	0.92	0.88	0.90
Van	0.90	0.90	0.90
Weighted Average	**0.87**	**0.86**	**0.86**

The model achieved an overall accuracy of **86.33%**. The highest performance was observed in the 'Pickup' and 'Truck' categories, with F1-scores of 0.91 and 0.90 respectively. These vehicles typically possess distinct visual features (e.g., cargo beds) that make them easier to distinguish. Conversely, the 'Crossover' class yielded the lowest F1-score (0.83), likely due to its visual similarity to SUVs (categorized under Offroad) and Hatchbacks.

B. Confusion Matrix Analysis

To better understand the misclassifications, we analyzed the confusion matrix shown in Figure 2.

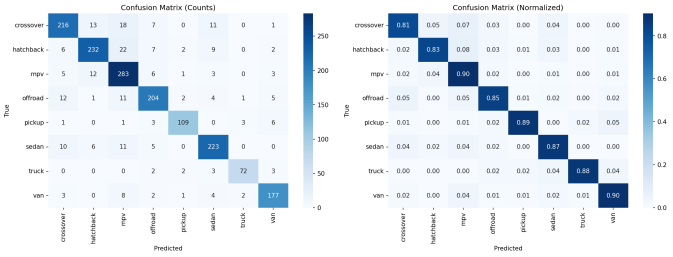

Fig. 2. Confusion Matrix of the Car Type Classifier on the Test Set. The diagonal elements represent correct predictions, while off-diagonal elements show misclassifications.

The confusion matrix reveals specific patterns of error:

- **MPV vs. Others**: While MPVs have a high recall (0.90), their precision is lower (0.80). The matrix indicates that other vehicle types, particularly Crossovers and Vans, are occasionally misclassified as MPVs. This is consistent with the visual ambiguity between these classes, as modern MPVs often share design traits with crossovers.
- **Crossover Confusion**: A notable number of Crossover vehicles are misclassified as MPVs or Hatchbacks. This supports the hypothesis that the definition of a 'Crossover' often overlaps visually with these other categories.
- **Sedan Robustness**: Sedans show balanced performance (0.87 Precision and Recall), indicating the model has learned robust features for this common vehicle type despite the high intra-class variance in sedan designs.

C. Per-Class Metrics Visualization

Figure 3 provides a visual comparison of Precision, Recall, and F1-Score across all classes.

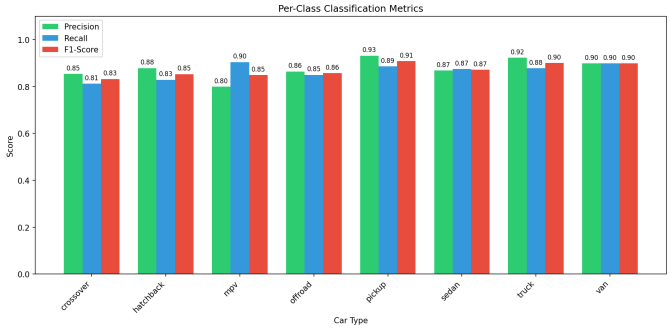

Fig. 3. Per-Class Classification Metrics (Precision, Recall, F1-Score) demonstrating consistent performance across categories.

The visualization confirms that the model maintains a high standard of performance across the board, with no single class suffering from catastrophic failure. This consistency is crucial for a reliable retrieval system.

D. Video Inference Evaluation

The full integrated pipeline was tested on a real-world traffic video (`traffic_test.mp4`, 197 seconds). The system processed 2,960 frames and generated the following statistics:

- **Throughput**: Average processing speed of 13.6 FPS. While not fully real-time (often defined as 30 FPS), this speed is sufficient for many surveillance and offline analysis applications.
- **Latency**: Average inference time per frame was 73.7ms used a standard GPU environment.
- **Detections**: A total of 14,107 car instances were detected and classified.
- **Distribution**: The detected vehicle distribution was: MPV (34.7%), Sedan (19.5%), Hatchback (16.1%), Offroad (12.0%), Van (9.0%), Pickup (4.1%), Crossover (3.1%), Truck (1.5%). This aligns well with the expected distribution of vehicles on Indonesian roads, predominantly MPVs and city cars.

V. CONCLUSION AND FUTURE WORK

We have successfully developed and validated a comprehensive Car Retrieval System tailored for the Indonesian context. By integrating a lightweight YOLOv8 detector with a robust ResNet50 classifier, we achieved an optimal balance between accuracy (86.33%) and inference speed (13.6 FPS). The detailed error analysis highlighting confusion between visually similar classes like MPVs and Crossovers provides clear directions for future improvements.

A. Future Work

Several avenues for future research and improvement are identified:

- **Fine-grained Classification**: Exploring Vision Transformers (ViT) or attention-based mechanisms to better distinguish between visually similar classes by focusing on specific discriminate parts of the vehicle (e.g., headlights, grilles).
- **Dataset Expansion**: augmenting the dataset with more samples of underrepresented classes like Trucks and Pickups to address class imbalance.
- **Edge Deployment**: Optimizing the models using quantization and pruning techniques (e.g., TensorRT, ONNX Runtime) to enable deployment on edge devices like NVIDIA Jetson for decentralized traffic monitoring.
- **Temporal Consistency**: Implementing tracking algorithms (e.g., DeepSORT) to smooth classifications over multiple video frames, reducing flickering predictions for the same vehicle instance.

ACKNOWLEDGMENT

The authors would like to thank the providers of the Indonesian car image dataset for making this research possible.

REFERENCES

[1] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE CVPR*, 2009, pp. 248–255.

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