# Vehicle Make and Model Recognition from Low-Detail Bangkok Metropolitan Administration CCTV Footage


Chutikarn Kamsen*, Nagul Cooharojananone*
*Department of Mathematics and Computer Sciences, Faculty of Science*
*Chulalongkorn University*
Bangkok, Thailand
chutikarn.ks03@gmail.com
Nagul.C@chula.ac.th

Supatana Auethavekiat
*Department of Electrical Engineering*
*Faculty of Engineering*
*Chulalongkorn University*
Bangkok, Thailand
Supatana.A@chula.ac.th

Pin Pin Tea-makorn†
*Data Analytics Center*
*Sasin School of Management*
*Bangkok, Thailand*
pinnaree.tea-makorn@sasin.edu

Pavitra Jindahra
*Data Analytics Center*
*Sasin School of Management*
*Bangkok, Thailand*
pavitraj@sasin.edu

Stefano Starita
*Data Analytics Center*
*Sasin School of Management*
*Bangkok, Thailand*
stefano.starita@sasin.edu

Kanokwan Atchariyachanvanich
*School of Information Technology*
*King Mongkut's Institute of Technology Ladkrabang*
Bangkok, Thailand
kanokwan@it.kmitl.ac.th



***Abstract*— In this paper, we introduce a deep learning-based system specifically engineered for the recognition of vehicle makes and models under challenging conditions, including low-resolution image and variable lighting scenarios. To achieve efficient object detection, we employed the You Only Look Once (YOLO) architecture, which is renowned for its ability to quickly process images while maintaining high accuracy. Our system is adept at detecting full-body vehicles captured in surveillance footage from Bangkok Metropolitan Administration CCTV cameras. The current implementation includes well-known models such as the Honda City, Civic, and Accord, as well as the Toyota Vios, Camry, and Corolla Altis (Altis). To evaluate the system's effectiveness, we conducted experiments across multiple lanes, simulating various distances and viewpoints typical of real-world surveillance situations. Zone B lane 2 achieved the highest precision of 100% with F1-score of 0.99. Other zones also exhibited commendable performance metrics, achieving F1-scores between 0.90 and 0.96. These findings illustrate the robustness of our system and its considerable potential for practical applications.**



***Keywords—CCTV Footage, Vehicle Model Recognition, Vehicle Make Recognition, YOLO***


## I. INTRODUCTION

Closed-circuit television (CCTV) systems have become increasingly vital for traffic monitoring, law enforcement, and public safety in metropolitan areas. In cities like Bangkok, surveillance cameras are widely deployed to track suspicious vehicles or reconstruct events after incidents. However, many of these systems suffer from limitations such as low resolution, poor lighting, and non-ideal camera angles, which significantly hinder the performance of vehicle recognition systems.

One of the key challenges in vehicle make and model recognition (VMMR) lies in insufficient image quality. CCTV footage often lacks visible key features such as logos, headlights, and body contours that are crucial for accurate classification. As a result, traditional methods that rely on high-resolution, front-facing images or license plates are rendered ineffective in these contexts.

Recent studies have explored various approaches to improve VMMR performance under adverse conditions. For instance, Chen et al. [1] proposed a method that combines PCA-NET with SVM to identify vehicles using frontal features and license plate positions, while Ren and Lan [2] leveraged convolutional neural networks (CNNs) to achieve accurate classification from frontal views. Mustafa and Karabatak [3] employed MobileNetV2 and YOLOx to handle low-resolution images captured in real-world scenarios, and Lyu et al. [4] introduced a two-branch, two-stage deep learning model to improve fine-grained recognition. Despite their effectiveness, most of these methods focus on frontal or high-quality images and are not optimized for low-resolution, full-body vehicle views commonly found in typical CCTV systems.

To address this gap, we propose an algorithm that performs make and model recognition of vehicles using full-body, low-resolution images extracted from real-world CCTV footage. Our method employs the YOLOv8 architecture for efficient vehicle detection and integrates brand-specific classifiers to distinguish between popular sedan models from Honda and Toyota. The system is designed to be robust to lighting variations and oblique camera angles and is evaluated on data collected from multiple city surveillance zones.

The main contributions of this work are as follows:

1) We develop a deep learning-based system tailored to identify vehicle make and model from low-resolution, full-body images captured in real surveillance conditions.

2) We build a custom dataset with bounding box annotations for sedans from Honda and Toyota brands, including models such as City, Civic, Accord, Vios, Altis, and Camry.

3) We demonstrate the effectiveness of our approach in multiple camera zones, achieving up to 95% accuracy and


**Chutikarn Kamsen and Nagul Cooharojananone are co-first authors.*
*† Pin Pin Tea-makorn is the corresponding author.*

average F1-scores above 0.90 in favorable zones, while maintaining competitive performance in challenging conditions.

The remainder of this paper is organized as follows: Section II reviews related work in vehicle recognition and low-resolution image processing. Section III describes the dataset and preprocessing. Section IV presents the proposed method and architecture. Section V explains the experimental setup, followed by the results in Section VI. Finally, Section VII concludes the paper and suggests directions for future research.

## II. Related Work

In recent years, VMMR has attracted growing interest due to its importance in intelligent transportation systems, traffic surveillance, and public safety. Traditional VMMR approaches initially relied on hand-crafted features, such as vehicle shape or grille geometry, and employed classical classifiers like Support Vector Machines (SVMs) [1] However, these methods often suffered from limitations under real-world conditions, such as varying viewpoints, occlusions, and low image resolution from CCTV cameras.

With the advent of deep learning, convolutional neural networks (CNNs) have revolutionized VMMR tasks. Ren and Lan [2] were among the first to demonstrate the effectiveness of CNNs in model recognition from cropped vehicle images. Hassan et al. [5] compared various CNN architectures including VGG16, ResNet, and MobileNet, finding that a fine-tuned VGG16 achieved over 99% accuracy, while lighter models offered faster inference with reasonable performance.

Recent studies have also adopted one-stage detectors such as YOLO. Ünal et al. [6] implemented YOLOv8 to jointly detect and classify vehicles into make and model classes in real-time. Their model achieved 94.3% accuracy and effectively processed low-resolution images from CCTV. To handle class imbalance and visual similarity between models, the authors emphasized dataset diversity and preprocessing techniques.

In contrast, two-stage pipelines have been explored to increase flexibility and robustness. Tsai et al. [7] employed Faster R-CNN for vehicle detection and a separate CNN for make and model classification, allowing for tailored optimization at each stage. Similarly, Lyu et al. [4] proposed a two-branch, two-stage (2B–2S) deep learning approach where one branch specialized in make classification and the other in model recognition.

Other approaches focus on specific vehicle parts to boost performance. Sultan et al. [8] used a three-stage system incorporating IWPOD-NET for logo alignment, YOLOv5 for logo detection, and EfficientNet for brand classification. This technique proved effective even under perspective distortion and small logos, though it could not distinguish vehicle models.

To improve generalization in complex scenes, attention mechanisms and ensemble learning have also been explored. Amirkhani et al. [9] combined visual attention with a multi-agent system and voting ensemble to improve accuracy under poor lighting and occlusions.

In summary, while early VMMR approaches relied on manual feature engineering, recent methods have shifted towards CNN-based and end-to-end deep learning pipelines. Among these, YOLOv8 stands out for its balance of accuracy and speed, especially under real-time and low-resolution conditions. Inspired by Lyu et al. [4] and Ünal et al. [6], this work adopts a two-stage pipeline using YOLOv8 for vehicle detection and separate models for make-specific classification (Toyota, Honda), addressing challenges such as class imbalance through data augmentation and preprocessing.

TABLE I. The Total Number Of Vehicle Images Per Class

| No. | Class Label | Number of images |
|---|---|---|
| 1 | Toyota Collora Altis | 3004 |
| 2 | Toyota Camry | 1338 |
| 3 | Toyota Vios | 878 |
| 4 | Honda Accord | 540 |
| 5 | Honda City | 886 |
| 6 | Honda Civic | 793 |
| 7 | Other (Benz,BMW,Nissan and other) | 1844 |

## III. Dataset and Pre-processing

To develop and evaluate our VMMR system, we constructed a custom dataset using CCTV footage collected from the Ratchathewi district in Bangkok, Thailand. The footage was captured from a single fixed-angle camera, offering a top-down perspective that included clear views of the vehicle front, side, and roof. All selected footage was recorded during daylight to ensure sufficient visibility of vehicle features.

The dataset focuses on six popular sedan models from two major brands Toyota (Vios, Altis, Camry) and Honda (City, Civic, Accord) as well as samples from other brands (e.g., Benz, BMW, Nissan) used for training a negative class. In total, 9,283 vehicle images were extracted and annotated with make and model labels, as summarized in Table I.

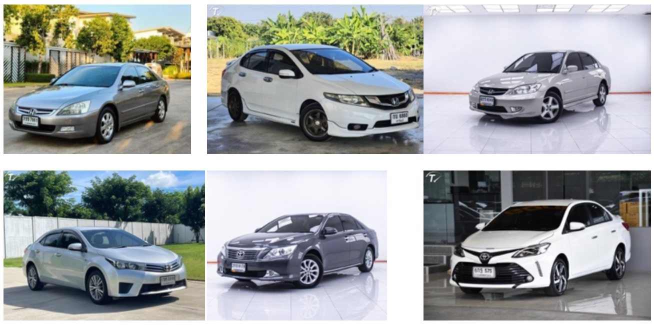

Fig. 1. Sample cars included in the dataset. Make and model clockwise from top left: Honda Accord, Honda City, Honda Civic, Toyota Corolla Altis, Toyota Camry, and Toyota Vios.

### A. Data Collection and Annotation

The dataset was compiled from 10 consecutive days of video footage, with 10 video clips recorded per day. Only clips with suitable lighting and minimal occlusion were selected. Vehicles were detected using a pretrained YOLO model with a confidence threshold of 0.9, and bounding boxes were used to crop the vehicle regions. These cropped images were resized to 640×640 pixels using stretching.

To improve recognition granularity, each car model was further divided into generation ranges based on manufacturing years. Examples include:

- Honda Accord: 2003–2008, 2008–2013, 2013–2019, 2019–2023
- Toyota Corolla Altis: 2001–2007, 2008–2013, 2014–2019, 2019–2024

All annotations were performed manually using the Roboflow tool, tagging each image with the corresponding make, model, and generation range. For instance, Honda Accord was subdivided into generation labels such as 2003–2008, 2008–2013, 2013–2019, and 2019–2023 (see Fig. 1). Labeling decisions were based on visual features such as headlights, grille shape, and body lines.

## B. *Dataset Distribution*

The dataset was divided based on capture dates, with 7 days of video used for training and validation, and the remaining 3 days for testing. Within the training-validation portion, 80% of the images were used for training and 20% for validation.

## C. *Pre-processing Pipeline*

*1) Frame Extraction and Filtering:* Video clips (10 clips/day for 10 days) were manually inspected and filtered to retain clear vehicle views. A pretrained YOLO object detector was applied with a high confidence threshold (0.9) to detect cars.

*2) Cropping:* Detected vehicle bounding boxes were cropped to eliminate background clutter.

*3) Resizing and Normalization:* Cropped images were resized to 640×640 pixels to match the input size for YOLOv8-based sub-models. Pixel values were normalized to [0, 1].

*4) Labeling:* Images were annotated using tools such as Roboflow, where bounding boxes and model-generation labels were manually applied. Labeling was guided by visual cues (e.g., grille shape, headlights) and chosen from three distance views: far, mid, and close (Fig. 8).

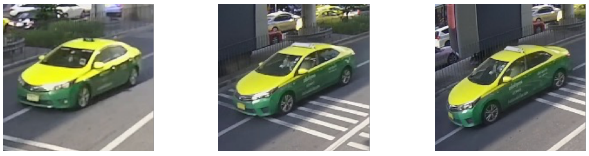

Fig. 2. Vehicle images at different distances used for labeling, consisting of three ranges: far view, mid view, and close view, respectively.

## D. *Handling Class Imbalance*

To address class imbalance, two techniques were applied:

- Augmentation: For minority classes, new samples were generated via horizontal flipping, rotation, brightness changes, and stretching. For example, Honda Accord had only 540 original images but was augmented to reach 2000 samples (see Fig. 10).
- Downsampling: Over-represented classes (e.g., Altis 2016–2019) were reduced to a maximum cap (e.g., 330 images per generation) to maintain balance across classes.

## E. *Labeling Strategy*

To enhance the model's discriminative capability, the dataset was divided into positive samples and negative samples for each make-specific sub-model.

TABLE II. SAMPLE GENERATION FOR TOYOTA COROLLA ALTIS BY GENERATION

| **Generation** | **Year** | **Total (images)** | **Augment** | **Down sampling** |
|---|---|---|---|---|
| 9th | 2001-2007 | 65 | 330 | |
| 10th | 2008-2010 | 105 | 330 | |
| 10th | 2010-2013 | 507 | | 330 |
| 11th | 2014-2016 | 705 | | 330 |
| 11th | 2016-2019 | 1095 | | 330 |
| 12th | 2019-2024 | 527 | | 330 |
| Total | | 3004 | 1980 | |

TABLE III. THE TOTAL NUMBER OF VEHICLE IMAGES AFTER BALANCING PER CLASS

| **Model** | **Old images** | **New images** |
|---|---|---|
| Toyota Collora Altis | 3004 | 1980 |
| Toyota Camry | 1338 | 2000 |
| Toyota Vios | 878 | 2000 |
| Honda Accord | 540 | 2000 |
| Honda City | 886 | 1980 |
| Honda Civic | 793 | 1950 |
| Other (Benz, BMW, Nissan and other) | 1844 | 1200 |

- Positive Samples: refer to images of vehicles that belong to the target classes (i.e., specific models of Honda and Toyota sedans). For instance, when training the sub-model responsible for classifying Honda vehicles, only the labeled images of Honda Accord, Honda City, and Honda Civic were treated as positive samples.
- Negative Samples: on the other hand, include images of vehicles not belonging to the make under consideration. For example, for the Honda-specific classifier, vehicles from Toyota, Benz, BMW, Nissan, and other non-Honda brands were labeled with the class other and used as negative training data.

This strategy helps the model learn not only to recognize the target vehicle models but also to reject unrelated vehicle types that may have similar visual characteristics. Including negative samples is particularly effective in reducing false positives, especially in real-world scenarios where vehicles from various manufacturers may appear similar in low-resolution CCTV footage.

Fig. 3. Illustrates some examples of vehicles used as negative samples for the Honda model classification.

# IV. PROPOSED FRAMEWORK

The proposed framework is designed to recognize vehicle make and model from low-resolution CCTV footage using a two-stage architecture. The system addresses practical challenges such as limited image resolution, non-frontal viewpoints, and variations in lighting. The framework integrates vehicle detection with YOLOv8 and make-specific classification models for Toyota and Honda sedans. The workflow is summarized in Fig. 12.

## A. *Stage 1: Vehicle Detection with YOLOv8*

The first stage employs YOLOv8 (You Only Look Once version 8) as the detection module. Trained on low-resolution samples, YOLOv8 provides both high accuracy and speed, making it suitable for real-time surveillance tasks. For each

CCTV frame, YOLOv8 detects vehicles belonging to the "car" class with a confidence threshold of 0.5. Bounding boxes around detected vehicles are expanded by 20% to ensure that essential features such as headlights, grilles, and contours are fully captured. Frames without valid detections are discarded.

To filter out irrelevant samples, vehicle candidates with an area smaller than 62,500 pixels or larger than 280,000 pixels are excluded, since these cases correspond to cars that are too far or too close to the camera and may reduce classification accuracy.

### *B. Stage 2: Make-specific Model Classification*

Once a valid cropped vehicle image is obtained, the system performs make and model classification using a hierarchical approach. Two separate CNN-based classifiers were trained independently:

- **Toyota classifier:** Vios, Altis, Camry
- **Honda classifier:** City, Civic, Accord

The classification pipeline begins by attempting to match the image against Toyota models. If the sample is classified as "other" by the Toyota classifier, it is passed to the Honda classifier for further evaluation. Vehicles that cannot be recognized as belonging to either make are categorized into the "other" class, which includes brands outside the paper scope (e.g., Benz, BMW, Nissan).

This brand-specific design reduces the search space for each classification step, improving performance and robustness under low-detail conditions. The final output consists of the predicted make-model label (e.g., Honda Accord) and its confidence score.

### *C. System Workflow*

The complete workflow of the proposed framework is illustrated in Fig. 6. The process begins with raw CCTV footage, followed by frame extraction, vehicle detection, size filtering, and hierarchical make-specific model classification. The final output is a labeled image with bounding boxes, model name, and classification confidence.

**Workflow steps**:

*1)* Input: Raw CCTV footage.

*2)* Frame Extraction.

*3)* Vehicle Detection (YOLOv8, confidence ≥ 0.5).

*4)* Size Filtering (62,500 ≤ pixels ≤ 280,000) Frame Extraction.

*5)* Toyota Classifier → output Toyota model name.

*6)* If not Toyota → Honda Classifier → output Honda model name.

*7)* If neither → assign to "Other".

*8)* Output: Labeled vehicle image with prediction of make and model.

This structured design ensures scalability, efficiency, and adaptability to real-world low-resolution CCTV environment.

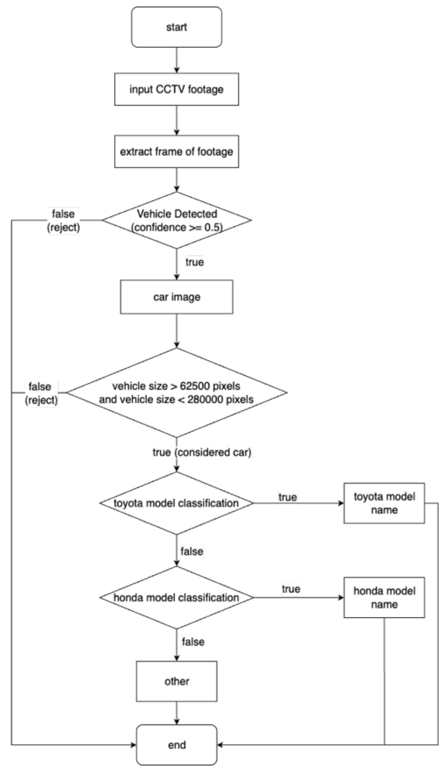


Fig. 4. The workflow diagram for our VMMR system.

## V. Experimental Setup

To evaluate the performance of the proposed system for VMMR from low-resolution CCTV footage, a series of experiments were conducted using real-world surveillance data. The experimental setup consists of the following components:

### *A. Hardware and Environment*

The experiments were conducted on a machine equipped with an NVIDIA RTX 3060 GPU (6GB VRAM), Intel Core i7 CPU, and 16GB RAM, running on Ubuntu 22.04. Python was used as the primary programming language, with support from PyTorch, OpenCV, and the Ultralytics YOLOv8 framework.

### *B. Dataset Preparation*

The dataset was constructed by extracting frames from low-resolution CCTV footage. Only front-view sedan vehicles were selected, focusing specifically on the two most common brands in the dataset: Toyota and Honda. Each frame was annotated with make and model labels, and resized to fit the YOLOv8 input size (typically 640×640 pixels). A total of 4 classes were defined for each sub-model: three specific models and one class labeled "other" to cover unmatched cases.

### *C. Model Architecture and Training*

The YOLOv8 model was employed as the detection and classification backbone. A two-step pipeline was used, where the first model detects vehicles, and the second classifies them into make and model. Two classification sub-models were trained separately for Toyota and Honda. Transfer learning with pretrained YOLOv8 weights was utilized, followed by fine-tuning on the custom dataset. Training was done over 100 epochs using a batch size of 16, with an Adam optimizer and learning rate set at 0.001.

### D. Data Augmentation and Preprocessing

To improve model generalization, various augmentation techniques were applied during training, including flipping, brightness adjustment, cropping, and noise addition. These techniques helped mitigate the effects of class imbalance and low image quality commonly found in CCTV footage.

### E. Evaluation Metrics

System performance was assessed using Precision, Recall, F1-score, and mean Average Precision (mAP). In addition, confusion matrices were generated to analyze class-wise performance and misclassifications.

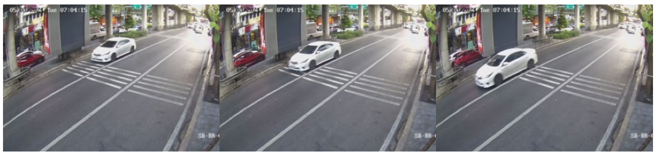

Fig. 5. Example images of vehicles in different positional zones: Zone A (before stop line), Zone B (aligned with stop line), and Zone C (after stop line).

Fig. 6. Example images of lane 2 (middle lane combined with lane adjacent to the sidewalk).

### F. Performance Evaluation of the recognition algorithm

The evaluation was conducted across two primary lanes:

- Lane 1: the lane adjacent to the center divider.
- Lane 2: The evaluation was conducted across two primary lanes: middle lane combined and lane adjacent to the sidewalk.

To capture positional variation, each lane was further divided into three zones:

- Zone A: Vehicle before crossing the stop line (far view)
- Zone B: Vehicle aligned with the stop line (mid view)
- Zone C: Vehicle past the stop line (close view)

## VI. RESULTS

This section reports the performance of the proposed VMMR system on low-resolution CCTV footage. We evaluate precision, recall, F1-score, confusion matrices, and report results by lane–zone as defined in Section V-F (Lane 1: adjacent to center divider; Lane 2: middle and sidewalk lanes).

### A. Lane–Zone Performance

Table III summarizes the results per zone and lane. The macro averages across all settings are Precision 94.58%, Recall 93.47%, and F1-score 93.99%.

TABLE IV. PRECISION/RECALL/F1-SCORE BY LANE AND ZONE

| Setting | *Precision (%)* | *Recall (%)* | *F1-score (%)* |
|---|---|---|---|
| Zone A lane 1 | 95.78 | 95.78 | 95.78 |
| Zone A lane 2 | 95.69 | 95.22 | 95.59 |
| Zone B lane 1 | 92.54 | 91.85 | 92.18 |
| Zone B lane 2 | 100.00 | 97.43 | 98.66 |
| Zone C lane 1 | 91.84 | 90.74 | 91.22 |
| Zone C lane 2 | 91.60 | 89.82 | 90.60 |
| Average | 94.58 | 93.47 | 93.99 |

Lane–zone results indicate the best performance in Zone B, Lane 2, consistent with vehicles being closest to the optimal viewpoint.

### B. Overall Performance

Table IV reports overall metrics computed on the test set, confirming the effectiveness of the two-stage design (YOLOv8 detector and make-specific classifiers).

TABLE V. OVERALL EVALUATION MERICS

| Metric | Value (%) |
|---|---|
| Precision | 98.40 |
| Recall | 93.35 |
| F1-score | 95.80 |

### C. Per-Class Evaluation

Per-class scores for the six target models are shown in Table V. Toyota and Honda models achieve consistently high precision and recall; the *Other* class is used only for analysis and is excluded from this table

TABLE VI. PER-MODEL PRECISION, RECALL, AND F1-SCORE.

| Model | *Precision (%)* | *Recall (%)* | *F1-score (%)* |
|---|---|---|---|
| Toyota Corolla | 98.25 | 95.08 | 96.60 |
| Toyota Camry | 99.58 | 94.50 | 96.70 |
| Toyota Vios | 97.50 | 94.08 | 95.99 |
| Honda Civic | 100.00 | 85.71 | 96.60 |
| Honda Accord | 94.12 | 85.33 | 89.48 |
| Honda City | 98.52 | 92.12 | 95.24 |
| Average | 98.40 | 93.35 | 95.80 |

### D. Confusion Matrix

Table VII presents the confusion matrix for the full test set. Most errors arise among visually similar models within the same make, while cross-make confusion is comparatively rare.

## VII. DISCUSSION AND ABLATION STUDY

This section presents a discussion of the key findings, limitations, and ablation results of the proposed system. Several observations were made during experimentation that provide insights into the behavior and robustness of the model.

### A. Effect of Camera Zones on Performance

The model's performance varied across camera zones due to differences in viewpoint, lighting conditions, and background clutter. As shown in Figure 15, zones with frontal or clear side views such as Zone A and Zone B yielded higher F1-scores (above 90%), while performance degraded in zone with oblique angles or occlusion. This highlights the importance of camera placement and view quality in CCTV-based vehicle recognition systems.

TABLE VII. CONFUSION MATRIX FOR THE FULL TEST SET

| | | predict | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | accord | city | civic | altis | camry | vios | other | car | reject |
| actual | accord | 64 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 4 |
| | city | 0 | 200 | 0 | 5 | 1 | 2 | 5 | 4 | 0 |
| | civic | 3 | 0 | 114 | 2 | 0 | 0 | 9 | 2 | 3 |
| | altis | 1 | 1 | 0 | 898 | 0 | 1 | 7 | 23 | 14 |
| | camry | 0 | 0 | 0 | 4 | 237 | 0 | 5 | 4 | 2 |
| | vios | 0 | 2 | 0 | 5 | 0 | 143 | 2 | 0 | 0 |
| | other | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| | car | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| | reject | NA | NA | NA | NA | NA | NA | NA | NA | NA |

### B. *Class Imbalance and Data Augmentation*

An ablation experiment was conducted to assess the impact of class imbalance. When training without augmentation on minority classes (e.g., Honda Accord, Toyota Camry), the average F1-score dropped by approximately 7.3%. By applying targeted data augmentation (rotation, flipping, contrast adjustment), performance improved, and the standard deviation across classes decreased. This confirms that augmentation is essential in balancing limited real-world datasets and avoiding overfitting to the majority classes.

### C. *Comparison Between Unified vs. Make-specific Models*

To evaluate the effectiveness of using make-specific sub-models, we compared the two-stage approach (with Toyota/Honda classifiers) against a unified model trained on all classes. The unified classifier showed increased confusion between similar-looking models (e.g., Civic vs. Altis), leading to a 4.1% decrease in macro F1-score. In contrast, make-specific classification improved intra-make discrimination and allowed the model to specialize more effectively.

### D. *Failure Cases and Limitations*

The model occasionally misclassified vehicles from the “Other” class into one of the known model classes, particularly when vehicles shared similar body types (e.g., Nissan Sylphy vs. Toyota Altis). This limitation suggests a need for an out-of-distribution (OOD) rejection mechanism or confidence-based filtering in deployment scenarios. Moreover, the reliance on manually assigned make labels for training presents a limitation; given the low resolution of the images, human annotators are susceptible to errors during classification. Future work will investigate methods and processes designed to mitigate these inaccuracies and ensure a more rigorous labeling phase.

## VIII. Conclusion and Future Work

In this work, we proposed a two-stage VMMR system tailored for low-resolution CCTV footage. Unlike conventional systems that rely on high-resolution frontal images or license plates, our method utilizes full-body vehicle images captured under real-world surveillance conditions. By leveraging YOLOv8 for object detection and make-specific sub-models for classification, the system achieves high recognition accuracy even with limited visual detail. Experimental results show that the proposed framework is effective and robust across varying camera zones, with an average F1-score exceeding 90%. Our ablation study confirms the benefit of separating classifiers by make and using targeted data augmentation to mitigate class imbalance. The system also demonstrates real-time capability, supporting its practical use in intelligent transportation and surveillance applications.

However, the current pipeline relies on pre-annotated make labels to route images to the appropriate sub-model. Additionally, misclassifications involving visually similar vehicles outside the target classes remain a challenge.

For future work, we plan to:

- Integrate an automatic make recognition module to eliminate reliance on manual routing.
- Extend the dataset to include more vehicle brands and types, improving system scalability.
- Incorporate temporal information from video sequences to enhance robustness under occlusion and motion blur.
- Explore lightweight deployment on edge devices for real-time roadside surveillance.

By addressing these directions, we aim to develop a more generalizable and scalable VMMR system suitable for widespread deployment in smart city infrastructures.

## References


[1] J. Chen, "Vehicle Brand Classification Method Based on PCA-NET Under Complex Background," *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, Kuala Lumpur, Malaysia, 2020, pp. 445-449, doi: 10.1109/CISCE50729.2020.00100.

[2] Yongguo Ren and Shanzhen Lan, "Vehicle make and model recognition based on convolutional neural networks," *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2016, pp. 692-695, doi: 10.1109/ICSESS.2016.7883162.

[3] T. Mustafa and M. Karabatak, "Real Time Car Model and Plate Detection System by Using Deep Learning Architectures," in *IEEE Access*, vol. 12, pp. 107616-107630, 2024, doi: 10.1109/ACCESS.2024.3430857.

[4] Y. Lyu, I. Schiopu, B. Cornelis, and A. Munteanu, "Framework for vehicle make and model recognition: A new large-scale dataset and an efficient two-branch two-stage deep learning architecture," *Sensors*, vol. 22, no. 21, Art. no. 8439, Nov. 2022, doi: 10.3390/s22218439.

[5] M. Hassan, S. Mahmood, and F. Aadil, "An empirical analysis of deep learning architectures for vehicle make and model recognition," *IEEE Access*, vol. 9, pp. 91487-91499, 2021.

[6] Y. Ünal, M. Bolat, and M. N. Dudak, “Examining the performance of a deep learning model utilizing YOLOv8 for vehicle make and model classification,” Journal of Engineering Technology and Applied Sciences, vol. 9, no. 2, pp. 131–143, 2024.

[7] C.-C. Tsai, C.-K. Tseng, H.-C. Tang, and J.-I. Guo, "Vehicle detection and classification based on deep neural network for intelligent transportation applications," in *Proc. 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Honolulu, HI, USA, Nov. 12–15, 2018, pp. 1605–1608.

[8] Z. Sultan, M. U. Farooq, and R. H. Raza, “Improved vehicle logo detection and recognition for complex traffic environments using deep learning-based unwarping of extracted logo regions in varying angles,” *Digital Interaction and Machine Intelligence*, vol. 12, 2023.

[9] A. Amirkhani and A. H. Barshooi, “DeepCar 5.0: Vehicle make and model recognition under challenging conditions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 541–553, 2022.