AI

MDPI

Article

Automatic Vehicle Recognition: A Practical Approach with VMMR and VCR

Andrei Istrate *, Madalin-George Boboc, Daniel-Tiberius Hritcu , Florin Rastoceanu *, Constantin Grozea and Mihai Enache

Military Equipment and Technologies Research Agency, 16 Aeroportului Street, 077025 Clinceni, Romania; gboboc@acttm.ro (M.-G.B.); dhritcu@acttm.ro (D.-T.H.); cgrozea@acttm.ro (C.G.); menache@acttm.ro (M.E.)

* Correspondence: aistrate@acttm.ro (A.I.); frastoceanu@acttm.ro (F.R.); Tel.: +40-723350872 (A.I.); +40-727831560 (F.R.)

Abstract

Background: Automatic vehicle recognition has recently become an area of great interest, providing substantial support for multiple use cases, including law enforcement and surveillance applications. In real traffic conditions, where for various reasons license plate recognition is impossible or license plates are forged, alternative solutions are required to support human personnel in identifying vehicles used for illegal activities. In such cases, appearance-based approaches relying on vehicle make and model recognition (VMMR) and vehicle color recognition (VCR) can successfully complement license plate recognition. **Methods:** This research addresses appearance-based vehicle identification, in which VMMR and VCR rely on inherent visual cues such as body contours, stylistic details, and exterior color. In the first stage, vehicles passing through an intersection are detected, and essential visual characteristics are extracted for the two recognition tasks. The proposed system employs deep learning with semantic segmentation and data augmentation for color recognition, while histogram of oriented gradients (HOG) feature extraction combined with a support vector machine (SVM) classifier is used for make-model recognition. For the VCR task, five different neural network architectures are evaluated to identify the most effective solution. **Results:** The proposed system achieves an overall accuracy of 94.89% for vehicle make and model recognition. For vehicle color recognition, the best-performing models obtain a Top-1 accuracy of 94.17% and a Top-2 accuracy of 98.41%, demonstrating strong robustness under real-world traffic conditions. **Conclusions:** The experimental results show that the proposed automatic vehicle recognition system provides an efficient and reliable solution for appearance-based vehicle identification. By combining region-tailored data, segmentation-guided processing, and complementary recognition strategies, the system effectively supports real-world surveillance and law-enforcement scenarios where license plate recognition alone is insufficient.

Keywords: vehicle make and model recognition (VMMR); vehicle color recognition (VCR); automatic vehicle recognition (AVR); real-world vehicle dataset; histogram of gradients (HOG); intelligent transportation systems (ITS); artificial intelligence (AI); deep learning; multitask strategy; support vector machines (SVM)

check for updates

Academic Editors: Kenji Suzuki and Miguel Angel Cazorla

Received: 15 October 2025
Revised: 29 November 2025
Accepted: 16 December 2025
Published: 18 December 2025

Citation: Istrate, A.; Boboc, M.-G.; Hritcu, D.-T.; Rastoceanu, F.; Grozea, C.; Enache, M. Automatic Vehicle Recognition: A Practical Approach with VMMR and VCR. *AI* **2025**, *6*, 329. https://doi.org/10.3390/ai6120329

1. Introduction

Automatic Vehicle Recognition (AVR) is a rapidly evolving subfield of computer vision and intelligent transportation systems, that seeks to automatically detect, identify and classify vehicles from digital images or video streams. It plays a central role in modern transportation infrastructure, enabling applications in traffic monitoring, toll collection, urban planning, fleet management, public safety and military use cases [1].

Early approaches to AVR relied heavily on license plate recognition, which became the de facto standard for vehicle identification. While license plate systems are well established and widely deployed, they face significant limitations, including susceptibility to counterfeiting, occlusion, adverse weather, and poor image quality. As a result, license plates alone often fail to provide a complete or reliable basis for vehicle identification in complex, real-world scenarios.

To address these challenges, recent research has shifted toward appearance-based vehicle recognition. Unlike license plate recognition, which has been the conventional approach, appearance-based recognition relies on inherent vehicle traits such as body contours, stylistic details and exterior color. Within this context, two critical tasks have emerged: Vehicle Make and Model Recognition and Vehicle Color Recognition. Together, these methods provide complementary attributes that strengthen identification beyond license plates, allowing systems to track or classify vehicles even when plates are missing, manipulated, or unreadable.

These capabilities are particularly relevant in use cases such as locating stolen cars, controlling access to restricted zones, or supporting security investigations. By enhancing robustness and reducing dependency on plates, VMMR and VCR contribute to more dependable monitoring infrastructures.

Despite recent progress in deep learning, achieving consistently high accuracy in VMMR and VCR remains difficult. The vast diversity of vehicle designs, weather conditions during capture, different camera perspectives, occlusions, and poor image quality, all introduce significant challenges. Furthermore, up-to-date and comprehensive datasets are required for reliable make and model classification, yet acquiring and maintaining such resources is demanding. For VCR, variations in illumination, reflections, and environmental factors often distort color perception, producing misclassifications, while the absence of diverse, high-quality images under varied conditions further limits performance. Generic systems do not reflect local distributions of makes, models, and colors. The absence of regional adaptation reduces deployment reliability: a system trained on North American or Asian vehicles may fail on European traffic compositions. Consequently, performance degrades in real scenarios where region-specific vehicles predominate, highlighting the importance of specialized dataset design. Misidentification in such contexts can limit investigations or delay responses. At the same time, research into localized datasets fosters advances in transfer learning and domain adaptation, relevant to the broader computer vision community. Additionally, the development of large foundational models—trained on diverse and geographically varied image sources—may help overcome these limitations by offering more robust generalization capabilities across regions, imaging conditions, and vehicle distributions.

There are several recent works that attempt to resolve some of the difficulties that still hinder the field of VMMR and VCR. A converging strand of research treats VMMR and VCR jointly within integrated AVR systems rather than as isolated modules [2–4]. The common approach is to fuse multiple vehicle attributes—make, model, type and color into a single operational workflow. Taken together, these three works advance an integrated view of Automatic Vehicle Recognition by coupling fine-grained VMMR with VCR so multiple attributes can be cross-checked within one pipeline. They demonstrate that merging make/model with color cues improves consistency checks and investigative search, and begin to address unseen vehicles. Yet all three implicitly highlight deployment sensitivities:

datasets usage, color recognition issues and the trade-off between closed set and open set generalization.

Custom datasets are essential to reflect regional vehicle distributions, yet remain costly and labor-intensive to build. VMMR models struggle with the trade-off between closed-set accuracy and open-set generalization, while fine-grained recognition is informative but error-prone compared to more robust coarse-grained solutions. Color recognition remains sensitive to lighting, reflections, weather, and class imbalance. Collectively, these works underline the need for systems that integrate fine-grained recognition with robustness and adaptability to region-specific deployments.

To confront these challenges, this paper introduces a VMMR and VCR system trained on a custom, region-tailored dataset reflecting local vehicle diversity. The proposed solution demonstrates that region-specific datasets combined with robust deep architectures with targeted preprocessing lead to improved results. By addressing challenges such as lighting, weather, viewpoints, and inter-class similarity through segmentation, the approach ensures higher accuracy, reliability, and usability for surveillance and security-critical contexts.

In the first stage, vehicles are detected using a pre-trained CNN detector (YOLOv8, Ultralytics, version 8.1.47 [5]), and discriminative cues are extracted for two recognition branches. For color recognition, the system employs semantic segmentation using DeepLabv3 [6] implemented in PyTorch (version 2.9.1)—combined with targeted augmentation (resized crop, horizontal flip, rotation, affine transforms) to stabilize predictions across viewpoints, lighting, and rain. For make-model recognition, HOG [7] feature extraction plus an SVM classifier is used to capture fine-grained distinctions without incurring heavy runtime. The VCR branch is benchmarked across EfficientNetV2 [8], MobileNetV3 [9], ResNet-50 [10], and ViT-B/16 [11] backbones. These models were intentionally selected in their lighter and well-established forms, as the primary goal was to construct the dataset efficiently and obtain preliminary insights into system performance. In our traffic-monitoring scenario, YOLOv8 proved sufficient for accurately detecting vehicles, while DeepLabv3 reliably segmented the relevant vehicle parts without requiring extensive custom training. This allowed rapid dataset creation and expedited the evaluation of the downstream recognition tasks. Evaluation relies on a set of task-appropriate metrics that ensure a complete assessment of model performance:

- Top-1 accuracy;
- Top-2 accuracy;
- Per-class precision, recall, and F1-score;
- Cohen's κ coefficient.

Training is performed on a custom, region-tailored dataset reflecting local fleets, color palettes, camera geometries, and adverse conditions. The proposed method yields 94.89% overall accuracy for VMMR, while for VCR, it reaches 94.17% Top-1 accuracy and 98.41% Top-2 accuracy, indicating strong performance under surveillance-like constraints. The resulting AVR stack is modular and consists of the following components:

- YOLOv8-based vehicle detection;
- Segmentation-guided VCR;
- HOG feature extraction for shape descriptors;
- SVM-based vehicle make-model classification.

This modular design ensures timely and consistent outputs suitable for law-enforcement and security-critical contexts. By coupling region-specific data with robust deep architectures and segmentation-driven color estimation, the system improves accuracy and robustness where generic, non-localized models often fail.

The remainder of this paper is organized as follows: Section 2 reviews recent work in VMMR and VCR, summarizing major datasets, considering size, availability, annotations, strengths and limitations. Section 3 describes the proposed solution with its methodology, including dataset development, preprocessing, and model design. Section 4 shows the experimental results and discusses findings and limitations, while Section 5 concludes with a summary and outlines directions for future research.

2. Related Work and Dataset

2.1. Related Work

Recent research in Vehicle Make and Model Recognition (VMMR) and Vehicle Color Recognition (VCR) has moved from traditional computer-vision pipelines toward integrated deep-learning systems intended for real-world use. Collectively, studies [2–4], and [12–31] have begun to address fine-grained discrimination, domain adaptation, adverse weather, and open-set/generalization issues through multitask designs, lightweight architectures, expanded datasets, and interpretability-oriented components (e.g., attention and part-aware segmentation).

VMMR research has matured significantly over the past decade. A comprehensive survey summarizing the transition from handcrafted descriptors to deep-learning-based approaches, emphasizing the need for large, geographically diverse datasets and robust domain adaptation is provided in [12]. Among major contributions, the VMMRdb dataset introduced in [13] remains a landmark: it contains over 291 k images across 9170 make-model–year classes and, when paired with a fine-tuned ResNet-50, achieved strong recognition results under unconstrained conditions. However, imbalance and U.S.-centric data limit its generalization to other regions. Similar efforts improved annotation quality and hierarchical labeling. Others restructured the CompCars dataset [14] removing data leakage and inconsistent type labels, demonstrating that hierarchical prediction pipelines and refined cropping can improve top-1 accuracy to around 70% under fine-grained settings. A two-branch deep model trained on the European DVMM dataset was proposed in [15], treating make and model as related but independent tasks. Their framework achieved consistent improvements (top-1 > 90%) while reducing confusion among visually similar classes, though it adds complexity and remains a closed-set system.

Several studies have focused on enhancing deployability and regional adaptability. The transfer learning with MobileNetV2 and EfficientNetV2-B3 on Lithuanian traffic imagery, reporting 88–92% accuracy with low computational cost but reduced robustness under night-time or cross-dataset conditions is applied in [16]. Building on this line of work, a hybrid attention module into MobileNetV3, guiding the network toward fine-grained cues such as grilles and headlights is incorporated in [17]. Their fine-tuned model achieved accuracy comparable to heavier backbones (ResNet-50, DenseNet-121) while maintaining real-time inference on edge hardware. They further extended this approach through regularized fine-tuning, which improved inter-class separability and convergence speed on the Stanford Cars dataset while preserving efficiency.

In earlier research, the handcrafted descriptors (HOG, GIST) with SVM and Random Forest classifiers obtained over 85% accuracy and real-time performance on modest processors. Although less robust than CNN-based systems, this approach underscores practical trade-offs between accuracy and computational simplicity for enforcement scenarios, is combined in [18]. In [19], the proposed method further advanced detection accuracy through an ensemble of EfficientDet and YOLOv8, achieving balanced precision–recall trade-offs across multi-view traffic datasets. These developments highlight the continued push toward compact yet high-performing architectures that support deployment in real-time vehicle-analytics systems.

In parallel, VCR studies sought to improve robustness against illumination, viewpoint, and weather variations. JADAR, a unified network that performs image deraining and color prediction simultaneously, improving mean average precision (mAP) in rainy scenes without extra latency is introduced in [20]. In [21], semantic-segmentation masks of car parts were used to isolate relevant regions before estimating dominant color, surpassing 90% accuracy and mitigating background bias. The UFPR-VCR dataset in [22] extended evaluation to low-light and night conditions, where EfficientNetV2 and ViT models exhibited notable accuracy drops (up to 20 percentage points), underscoring the need for domain-specific adaptation. GAN-based augmentation methods such as ColorGAN [23] further expanded the available color diversity, improving YOLOv5 detector mAP by approximately 5 points without increasing inference cost. Overall, these findings demonstrate that combining segmentation, restoration, and synthetic-data enrichment yields greater gains in color recognition than deeper backbones alone.

Recent research increasingly integrates VMMR and VCR within multitask AVR pipelines, coupling make-model, color, and sometimes license-plate recognition into unified workflows. In [2,3], showed that such unified frameworks enhance cross-attribute validation and retrieval consistency, achieving > 90% accuracy across tasks. Kim [4] applied a multitask deep network to recognize vehicle type and color simultaneously, attaining approximately 94% color and 90% type accuracy, demonstrating the synergy between visual attributes. Recent extensions employ attention- or transformer-based models [24–27], multitask fusion, and synthetic pretraining on datasets such as Synset Boulevard [28] to address unseen vehicles and improve open-set robustness. Complementary analyses in [29] surveyed more than fifty deep-learning frameworks for vehicle detection and classification, emphasizing the convergence toward transformer-enhanced CNN hybrids and ensemble models for state-of-the-art accuracy while calling for benchmark standardization. Furthermore, modern backbone designs such as ResNeXt [30] and ConvNet/ConvNeXt [31] have influenced recent AVR architectures by introducing modular residual transformations and convolutional refinements that improve transfer learning and generalization on fine-grained automotive datasets. These integrated approaches represent the field's current direction—balancing accuracy, interpretability, and efficiency within practical, region-adapted recognition systems.

Despite notable progress, several challenges remain. Current models still struggle with domain shifts between web and surveillance imagery, low-visibility environments, and long-tail vehicle categories. Future research should emphasize the creation of large, balanced, and region-specific datasets, adaptive augmentation strategies for adverse conditions, and open-world benchmarking protocols to ensure transparent and reproducible evaluation of AVR systems in real-world deployments.

The evolution of these recognition methods has been closely tied to the availability of annotated datasets that capture the variability of real-world traffic scenes. As the surveyed literature demonstrates, the choice of dataset directly affects model generalization, especially across regions, lighting conditions, and viewpoints. Therefore, the next subsection reviews the main public and custom datasets used in VMMR and VCR research, outlining their scale, annotation schemes, and relevance for practical deployment.

2.2. Major Vehicle Recognition Datasets

The performance of recognition systems is greatly influenced by the quality of the datasets used for training. Table 1 presents the best datasets, highlighting their size, availability, annotation schemes, usefulness for VMMR and VCR, as well as their general strengths and limitations.

Table 1. VMMR and VCR datasets.

No.	Dataset (Name: Year)	Availability ● Images ● Classes	Annotation Type ● VMMR/ VCR Usage	Advantages/ Disadvantages	References
1	Stanford Cars (Cars196): 2013	Public (research mirrors) ● 16,185 images ● 196 classes (MMY—model, make, year)	Image-level make-model–year; devkit has bboxes. ● VMMR usage: fine-grained classification.	+ Clean, fine-grained labels; standard FGVC-fine-grained visual clasification benchmark. − Web images; limited surveillance viewpoints.	[2,3,12,17,24,25]
2	NTOU-MMR: 2014	Public (research) ● 6639 images (2846 train/3793 test) ● 29 classes	Image-level make-model (frontal surveillance). ● VMMR usage: traditional ML baselines and real-traffic testbed.	+ Early real-traffic set; simple splits. − Small; class imbalance; narrow views.	[12,18]
3	CompCars (Web + Surveillance): 2015	Non-commercial research license ● 136,726 web + 27,618 parts + 50,000 surveillance ● 1716 models (163 makes)	Web: bboxes/viewpoints/attributes; Surveillance: front-view labels. ● VMMR usage: large-scale fine-grained classification; hierarchy; parts.	+ Large, multi-scenario; rich attributes. − License restricted; web and surveillance domain gap; surveillance is front-view only.	[12,14,26,27]
4	VehicleID (PKU-Peking University): 2016	Public (registration/request) ● 221,763 images ● 26,267 IDs (subset with model labels)	Re-ID IDs; partial model labels. ● VMMR usage: limited; mainly Re-ID.	+ Large multi-camera Re-ID set. − Daytime bias; incomplete make/model labels for VMMR.	[12]
5	VeRi-776: 2016	Public (by request; non-commercial) ● ≈50 k images ● 776 IDs	IDs, bboxes, type/color/brand, plate and spatio-temporal meta. ● VCR/VMMR usage: attributes; Re-ID benchmarks.	+ Rich attributes and multi-view; widely benchmarked. − Moderate size; single-city domain.	[12]
6	BoxCars116k: 2017	Public ● 116,286 images ● 693 fine-grained classes	3D bbox/viewpoint ‘unpacking’; fine-grained make-model-submodel-year. ● VMMR usage: surveillance FGVC-fine-grained visual clasification.	+ Many viewpoints; strong for surveillance FGVC-fine-grained visual clasification. − Long-tail classes; limited non-appearance attrs.	[12]
7	VMMRdb: 2017	Public (MIT-Massachusetts Institute of Technology) ● 291,752 images ● 9170 classes (MMY—model, make, year)	Image-level make-model–year (no bbox). ● VMMR usage: very large class coverage across decades.	+ Huge coverage; fine-grained. − Quality/label noise; heavy imbalance.	[12,13]
8	VeRi-Wild: 2019	Public (request; non-commercial) ● 416,314 images ● 40,671 IDs	Re-ID IDs from 174 cameras over a month; vehicle crops. ● VCR/VMMR usage: attributes via Re-ID; domain for surveillance.	+ Very large; high variability (view/illumination/occlusion). − No make/model labels; imbalance across IDs.	[12]
9	MVP (Multi-grained Vehicle Parsing): 2020	Parsing annotations public (images from VeRi/AICITY19/VeRi-Wild) ● 24,000 images ● 10 coarse parts/59 fine parts	Pixel-level part masks for part-aware training. ● VMMR/VCR usage: parts-aware recognition and color estimation.	+ Enables part-aware VMMR/Re-ID; complements appearance cues. − Sourced from other sets; no make/model class labels.	[12]
10	DVMM: 2022	Public (by request) ● Large-scale VMMR dataset. 281,133 images, 326 models, 23 makes	Framework + dataset for VMMR; two-branch, two-stage deep learning. ● VMMR usage: fine-grained make/model.	+ Newer large-scale data; efficient two-branch/two-stage baseline. − Access/logistics may be by request; region bias possible.	[15]
11	Synset Boulevard: 2024	Public (synthetic-artificial generated images) ● 500,000 images ● 800 vehicle classes	Synthetic 3D renders with segmentation, pose, color metadata. ● VMMR/VCR usage: pretraining; domain randomization; rare-case coverage.	+ Perfect labels; controllable domains. − Synthetic-to-real gap; limited realism in adverse conditions.	[28]
12	GLPD (Global License Plate Dataset): 2024	Public (request) ● 1.2 M images ● Global fleet coverage	ALPR-centric with vehicle attributes (bboxes, plate text, make-model-color). ● VMMR/VCR usage: unified AVR (ANPR + VMMR + VCR) and cross-attribute checks.	+ Massive, multi-region coverage; rich labels. − Heavily ALPR-focused; quality varies; not solely VMMR/VCR.	[2]
13	UFPR-VCR: 2024	Academic license (request; free for non-commercial research) ● 10,039 images ● 11 colors (9502 unique vehicles)	Color labels; frontal and rear; includes night/occlusion. ● VCR usage: color classification under adverse/night conditions.	+ Hard VCR benchmark (night/adverse) with validated color labels. − Color-only task; sourced from Brazilian ALPR sets.	[22]
14	Car-1000: 2025	Public (research license) ● 240,000 images ● 1000 fine-grained classes	Image-level make-model–year; balanced web and traffic imagery. ● VMMR/VCR usage: balanced fine-grained VMMR; optional color benchmarking.	+ Balanced multi-region data; supports multitask VMMR/VCR. − Mixed lighting quality; limited rare-model coverage.	[12,17,24]

“●” denotes itemized dataset characteristics; “+” indicates advantages; “−” indicates limitations.

Curated vehicle datasets underpin progress in VMMR/VCR by standardizing evaluation, enabling fair comparisons, and exposing models to diverse conditions. Legacy sets (Cars196, NTOU-MMR) offer clean, fine-grained labels or early surveillance views, while CompCars, BoxCars116k, and VMMRdb scale class coverage and viewpoints for robust fine-grained learning. Re-ID collection (VehicleID, VeRi-776, VeRi-Wild) contribute multi-camera variability and rich metadata, and parsing resources enable part-aware modeling. Recent VCR/AVR datasets (UFPR-VCR, GLPD) add color labels, plates and global fleets; synthetic collections (Synset Boulevard [28]) deliver perfectly annotated variety; newer compendia (Car-1000) balance web/traffic imagery. Benefits include scale, attributes and realism; drawbacks include class imbalance, domain shift, licensing limits and regional bias.

3. Proposed Solution

In this section, we present our proposed method based on a hybrid system that integrates dedicated modules for both color detection and make-model recognition. The following subsections outline an overview of the methods employed, system architecture and the experimental procedures described in detail.

Machine learning techniques are not taught by design to focus on specific vehicle components when addressing the problem of vehicle identification. This contrasts with the way humans approach the same task: when asked to identify a car, we naturally rely on distinctive components of its body. We present the methods used in the color identification phase, where pixel masks are extracted for each relevant component of the vehicle body, while unnecessary background information is discarded. This targeted filtering ensures that the model focuses only on meaningful regions that contribute to reliable color classification.

For the color identification task, the image is processed so that only specific body parts are retained. Components such as wheels, lights, front grille, license plate, windshield, and lateral windows are excluded because they do not consistently reflect the vehicle's main body color.

As far as the model recognition task is concerned, the same extraction method is used but with a different focus. The most distinctive features are found on the front side, where design elements like the grille and shape of headlights provide strong features for identification. The lateral side is less informative, as its appearance is largely common across vehicles.

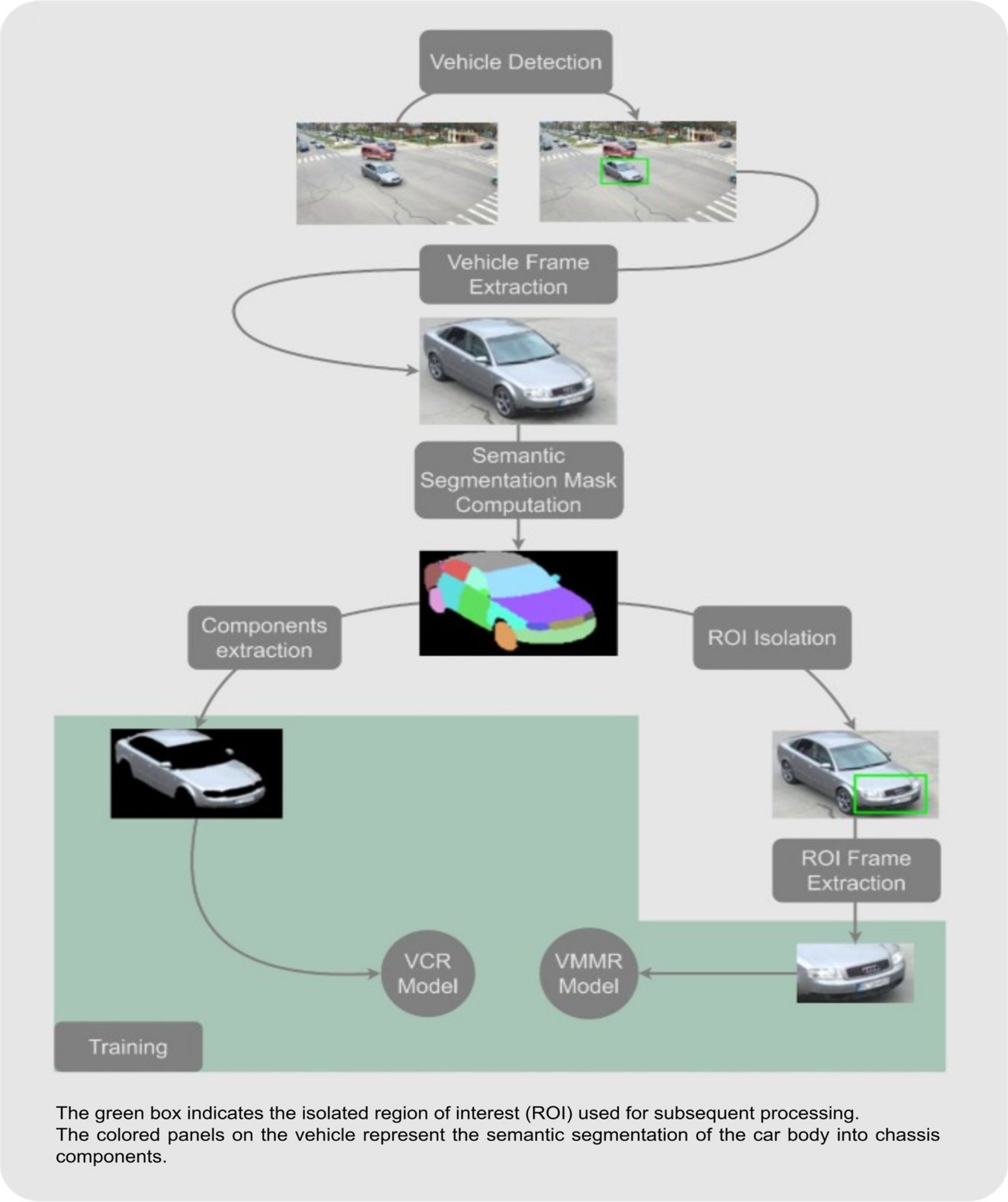
The second stage of the pipeline focuses on extracting fine-grained regions of interest from the detected vehicles. For this purpose, we employed a pretrained segmentation model due to its ability to capture multi-scale contextual information through spaced convolution and feature pyramid representations [6]. The model was trained on the MVP Dataset [32], which provides high-quality pixel-level annotations for vehicles under diverse angles and conditions.

The segmentation model is available in two versions:

- COARSE—creates partition with 9 components;
- FINE—creates partition with 59 components.

In our work, we adopted the fine version, since our task required isolating fixed regions. From each input vehicle image, the segmentation model generates a mask that separates different structural components of the car.

The overall architecture of the proposed system is illustrated in Figure 1. It shows the complete pipeline, starting from the acquisition of real-world vehicle images and going further with feature extraction, and model training.



The green box indicates the isolated region of interest (ROI) used for subsequent processing.
The colored panels on the vehicle represent the semantic segmentation of the car body into chassis components.

Figure 1. Proposed system architecture.

The proposed system architecture consists of two main components: the Detection Node, which ensures that the essential parts of the vehicle body are considered for accurate color prediction, and the Semantic Segmentation Node, which guarantees that only the pertinent regions of the vehicle are used for make-model recognition.

The vehicle detection node is responsible for identifying and isolating vehicles captured from real traffic images. This is achieved using the pre-trained CNN-based detection model [5], chosen for its efficiency and strong performance. Bounding-boxes are generated around detected vehicles with high confidence scores and a margin of 5–10% is taken into consideration. This ensures that areas such as the roof edges, bumpers or side contours are preserved, while minimizing background noise. The bounding-boxes are cropped to produce fixed-type input samples for further semantic segmentation processing.

The filtered mask is then resized to match the resolution of the original RGB image and applied to it, preserving only the regions of interest and discarding background pixels. During this operation, nearest-neighbor blending is applied when overlaying masks onto RGB images, ensuring that no interpolation artifacts or color mixing occur at the mask boundaries. This process ensures that subsequent classification models receive consistent inputs, its description being shown in Figures 2 and 3.

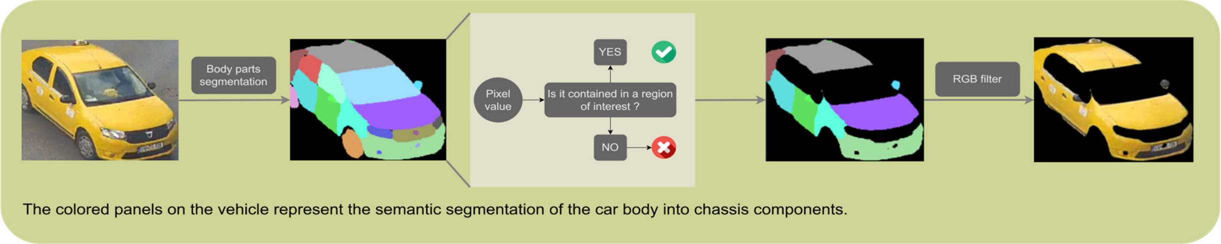

Body parts segmentation
Pixel value
Is it contained in a region of interest ?
YES
NO
RGB filter
The colored panels on the vehicle represent the semantic segmentation of the car body into chassis components.

Figure 2. Component extraction.

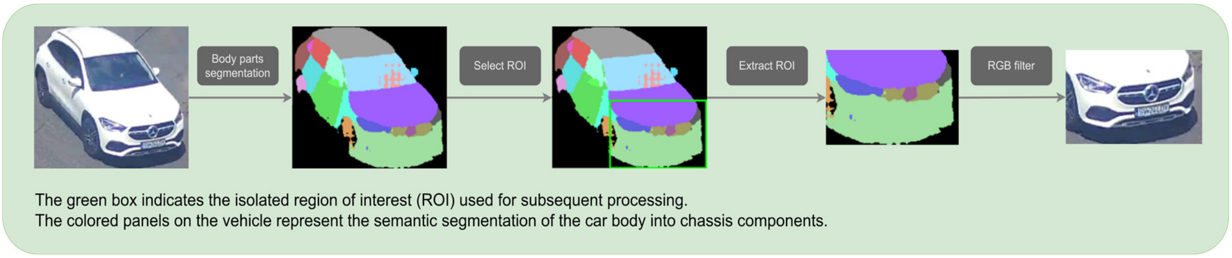

Body parts segmentation
Select ROI
Extract ROI
RGB filter
The green box indicates the isolated region of interest (ROI) used for subsequent processing.
The colored panels on the vehicle represent the semantic segmentation of the car body into chassis components.

Figure 3. ROI frame extraction.

For the color identification task, only the regions relevant to the vehicle's body color are retained from the segmentation mask. Specifically, the hood, the roof and lateral doors are meaningful, while all other parts are removed. This may be observed in Figure 2.

For the make-model recognition task, as shown in Figure 3, the segmentation masks are used to guide the extraction of a highly discriminative region of interest. By locating structural landmarks such as the front wheels and headlights, we isolate the area extending from the headlights down to the bottom of the front bumper. This region is particularly rich in stylistic features that differentiate one make-model-generation class from another, such as the grille design, headlight shape and bumper contours.

4. Experimental Workflow

To evaluate the proposed system, we designed a series of experiments that validate both the vehicle color identification and the make-model recognition modules. The workflow is structured into two parts. First, we describe the custom datasets created from real-time camera captures. Next, we present the experimental setup, which details the data processing pipeline, the training procedures and the model selection.

4.1. Datasets

The images used in our experiments were collected using real-world traffic footage acquired from cameras positioned at two different intersections. All images were manually annotated to ensure precise labeling for both the VMMR and VCR tasks and have been obtained following the methods illustrated in Figures 1–3. An additional critical aspect is that all photos were taken exclusively during daytime.

The datasets accurately reflect current European traffic compositions, encompassing a representative range of vehicle types (sedans, SUVs, hatchbacks, compact cars). Motorbikes, bicycles, and trucks were intentionally excluded because these vehicle categories often exhibit multiple or non-uniform characteristics that could distort color classification and compromise label consistency. The images used in our experiments were collected using real-world traffic footage acquired from cameras positioned at two different intersections. This multi-site collection strategy was crucial as we observed a different trend of car usage, make, and model distribution between the two regions, thereby ensuring greater local vehicle diversity. To further enhance robustness, data collection spanned two different seasons, introducing significant variance in illumination, shadows, and road conditions (dry vs. rainy/foggy surfaces). The captured footage reflects realistic traffic dynamics, including vehicles in motion at different speeds, and vehicles viewed from various camera perspectives, with challenging camera angles of approximately, sliding on the horizontal front side and on the vertical front side. This combination of diverse capture sites, seasonal lighting changes, and realistic traffic dynamics makes the resulting dataset highly representative of challenging, operational surveillance environments, providing a strong contribution to European research datasets as it is representative of current car models used in the area. While this collection can be improved with the addition of multiple vehicle classes to broaden coverage, its current structure provides sufficient and high-fidelity data for our fine-grained recognition tasks. Separate datasets were prepared for the color identification and make-model recognition tasks.

For the color identification task, the dataset contains images distributed across 8 color categories: silver, white, black, blue, red, gray, yellow, and green. The first version of the dataset (Figure 4) exposed 1748 images. Although it was sufficient in size, it suffered from class imbalance, with certain categories having significantly fewer samples compared to other classes (starting from blue).

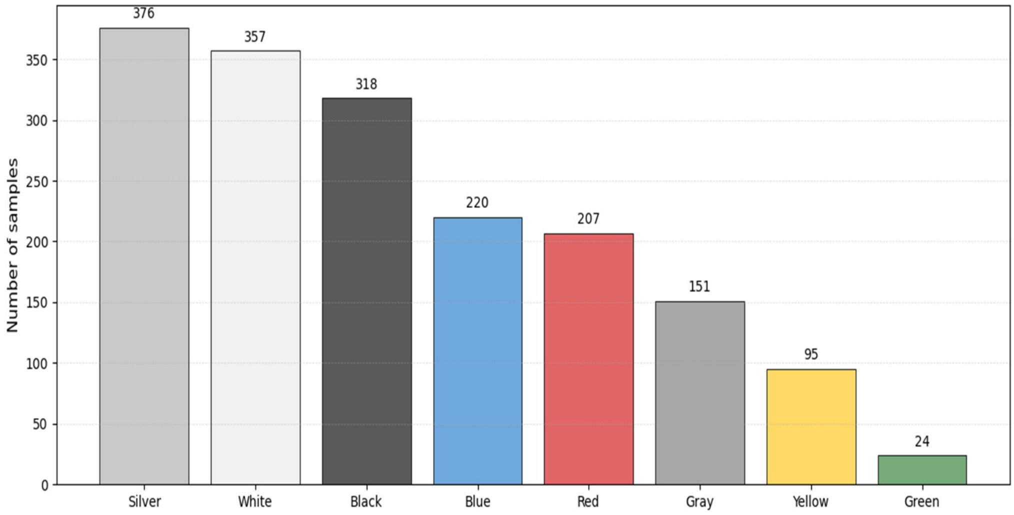


Figure 4. Initial color distribution.

To address this limitation, the initial dataset was adjusted (Figure 5). This contains 2343 images, obtained under the same conditions and covering similar color categories, but with a more uniform distribution across them. In this way, the bias toward dominant classes was reduced and the robustness of the color identification model was improved.

All images used to create both the color identification and make-model recognition datasets were captured in a realistic manner, bringing the experimental setup closer to real-life deployment scenarios. Several aspects contribute to this fact:

- Vehicles were recorded while in motion, with speeds ranging from 15 km/h to 70 km/h, reflecting actual traffic conditions;
- Data was collected under different weather conditions, including sunny, rainy and foggy;
- Images were captured during various periods of the day with the exception of nighttime;
- The camera angle of view was approximately 40 degrees on the horizontal front side and up to 60 degrees on the vertical front side.

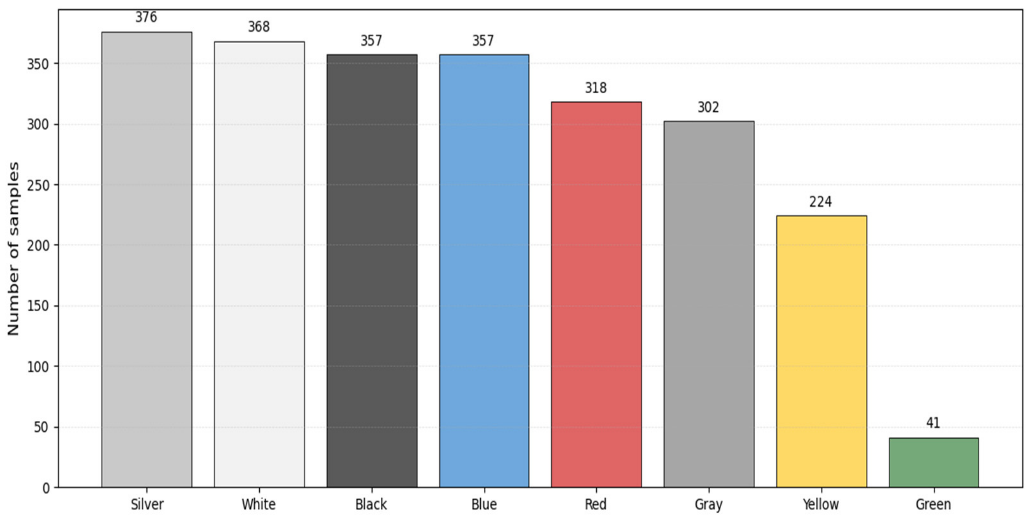


Figure 5. Updated color distribution.

As far as the make-model recognition task is concerned, we created a specific dataset using the same data source. It has 1955 images, divided into 24 classes, where a class refers to a specific make-model-generation combination. The dataset covers 7 manufacturers and a broad generation pool, ranging from 2000 to 2024. The details about selected classes are presented in Table 2.

Table 2. Vehicle make-model categories.

No.	Make-Model		Generation	Train	Test
1	BMW	5 Series	2009–2013	32	9
2	BMW	X3	2010–2013	36	10
3	Dacia	Logan, MCV, Sandero	2016–2020	184	46
4	Dacia	Duster	2013–2021	71	28
5	Dacia	Duster	2024	50	13
6	Dacia	Logan, MCV	2004–2012	107	27
7	Dacia	Logan, Sandero	2020–2024	125	32
8	Dacia	Spring	2021	33	9
9	Ford	Focus	2004–2008	39	10
10	Renault	Megane	2003–2009	36	10
11	Renault	Megane	2016–2020	39	10
12	Renault	Clio	2013–2019	37	10
13	Skoda	Octavia	2004–2008	24	7
14	Skoda	Octavia	2019–2024	26	7
15	Toyota	Auris	2013–2018	12	4
16	Volkswagen	Golf	2000–2003	126	32
17	Volkswagen	Golf	2008–2012	58	15
18	Volkswagen	Passat	2000–2005	68	17
19	Volkswagen	Passat	2005–2010	180	46
20	Volkswagen	Passat	2010–2014	55	14
21	Volkswagen	Passat	2014–2019	44	12
22	Volkswagen	Tiguan	2008–2011	77	20
23	Volkswagen	Touran	2003–2006	57	15
24	Volkswagen	Up	2012–2023	28	8

This dataset design ensures representation of both inter-class variability (across different make-model combinations) and intra-class difference (across different generations of the same model). For instance, the Volkswagen Passat spans four generations in the dataset, allowing the system to learn not only brand- and model-specific features but also the stylistic differences introduced over time (see Figure 6).

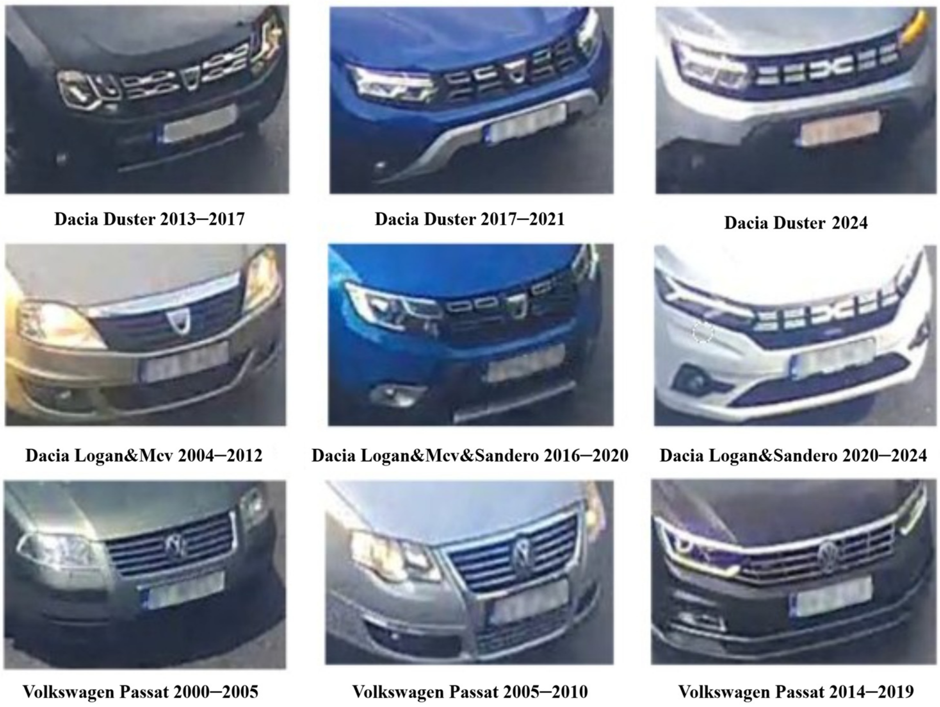

Dacia Duster 2013—2017
Dacia Duster 2017—2021
Dacia Duster 2024
Dacia Logan&Mcv 2004—2012
Dacia Logan&Mcv&Sandero 2016—2020
Dacia Logan&Sandero 2020—2024
Volkswagen Passat 2000—2005
Volkswagen Passat 2005—2010
Volkswagen Passat 2014—2019

Figure 6. Make-Model dataset samples.

4.2. Experimental Setup

All deep learning models were implemented in PyTorch (version 2.9.1) using Python (version 3.12), while the SVM classifier was implemented using Scikit-learn (version 1.6.1).

Both datasets have been split into training (80%) and testing (20%) subsets, in order to maximize the number of samples available during training while maintaining a sufficiently large and representative subset for evaluation. To ensure the reliability of the reported results, the experiment was conducted over three independent iterations of dataset splitting. For each iteration, models were trained and evaluated separately, and the final performance metrics represent the average of the results obtained across all three runs.

Regarding vehicle color classification, we evaluated 5 state-of-the-art architectures:

- EfficientNetV2, known for high accuracy with optimized parameter efficiency [8];
- MobileNetV3, designed for lightweight, fast inference on edge devices [9];
- ResNet50, a widely adopted residual network capable of deep feature extraction [10];
- ViT-B16, leveraging attention mechanisms to capture long-range dependencies in visual data [11];
- ConvNeXt, a modernized convolutional network inspired by transformer architectures, combining convolutional efficiency with design elements from Vision Transformers [33];

To enhance model robustness and prevent overfitting, we applied data augmentation techniques during preprocessing. These augmentation techniques are particularly beneficial for the vehicle color recognition task, as they simulate natural variations in vehicle positioning, orientation, and camera perspective commonly encountered in real-world traffic scenes. By exposing the model to such controlled geometric changes, the training process becomes more robust to viewpoint differences, ultimately enhancing classification accuracy and overall model generalization. The list of applied augmentation operations is detailed as follows:

- Resized Crop—we randomly crop a region of the image and then resize it to 224 × 224 pixels, keeping between 85% and 100% of the original input;
- Horizontal Flip—with a probability of 80% the image is flipped horizontally;
- Rotation—the image is randomly rotated by up to ±15 degrees;
- Affine Transformation—up to 10% translation, between 90% and 110% scaling, and up to ±15 degrees shear;

All the images have been normalized with the mean and standard deviation from ImageNet [34]. Training has been executed using the Adam optimizer [35] with an initial learning rate of 0.001. We employed a StepLR schedule that reduced the learning rate by a factor of 0.1 every 50 epochs. Models were trained in batches of 32 images for up to 500 epochs, with early stopping after 15 epochs of no improvement to avoid overfitting. In practice, no model trained for more than approximately 200 epochs, as the stopping condition was typically triggered accordingly.

When addressing the make-model classification task, we reused two of the augmentation techniques applied in the color classification task (Horizontal Flip and Rotation). In addition, we introduced methods better suited for the selected feature extraction technique:

- Gaussian Blur—with a 30% probability a slight blur is applied using a kernel size of 3 × 3 or 5 × 5;
- Brightness—with a probability of 70%, the image brightness is shifted by up to ±20 units;
- Contrast—with a probability of 70%, the image contrast is varied between 0.8 and 1.2;

For the training process, we combined a feature extraction approach with a linear classifier. The system uses HOG features as input to an SVM classifier. HOG features were extracted using OpenCV (version 4.12.0), and classification was performed using an SVM implemented in Scikit-learn (version 1.6.1). We chose this method as it is well-suited for smaller datasets, remains robust even when class frequencies are not uniform, and also offers greater control over the feature extraction and classification parameters. HOG descriptors were extracted using a grid of cells, each divided into 9 orientation bins. To improve robustness to illumination differences, gamma correction was applied, and the concatenated feature vectors were normalized using the L2 norm. Several HOG spatial configurations were evaluated by varying the block dimensions—(16, 5), (20, 8), (25, 9), (33, 9), and (40, 10)—combined with overlapping cells of size (2, 2). The extracted features were used to train an SVM classifier, where multiple kernel configurations were tested, including RBF with parameter values (1, 2, 4, 6, 8). A one-vs-one decision function was adopted, so binary classifiers were constructed for every pair of classes.

For both tasks, the training of the models was carried out on a 64-bit operating system workstation equipped with an Intel® Core™ i7-8565U CPU (Intel Corporation, Santa Clara, CA, USA), NVIDIA GeForce GTX 1050 GPU (NVIDIA Corporation, Santa Clara, CA, USA) and 8 GB of RAM.

4.3. Results

In this section the results are reported, highlighting the performance of each module and discussing the effectiveness of the proposed approach.

As far as the evaluation metrics are concerned, these were presented independently for the two tasks, based on the nature and goals of each. For the color classification task, we report accuracy per class values (Top-1 and Top-2) and confusion matrix, so that we can quantify prediction correctness and error distribution. Precision, recall and f1-score per class values are illustrated for the make-model classification task, where class imbalance and inter-class similarity requires metrics that emphasize not only correct predictions but also the method's ability to detect each class consistently.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Top-1 Accuracy} = \frac{\text{Number of samples where the highest probability prediction is correct}}{\text{Total number of samples}} \tag{5}$$

$$\text{Top-N Accuracy} = \frac{\text{Number of samples where the correct class is within the top N predicted probabilities}}{\text{Total number of samples}} \tag{6}$$

As it can be observed in Table 3, all the models achieved strong performance on Top-1 results, with EfficientNetV2 and ResNet50 obtaining the highest weighted accuracy. ConvNeXt also demonstrated consistently strong results overall. However, similar to ViT-B16, it underperformed on "Green" unbalanced class. This behavior is consistent with its architectural design, inheriting comparable sensitivity to class imbalance in fine-grained visual tasks.

Table 3. Color classification accuracy (Top-1).

Model	White	Blue	Yellow	Silver	Gray	Black	Red	Green	Weighted Acc	Avg. Acc
EfficientNetV2	100	95.08	97.78	100	79.69	91.89	100	88.89	94.93	94.17
MobileNetV3	94.44	96.72	97.78	92.11	95.31	86.49	100	77.78	94.08	92.58
ViT-B16	100	98.36	97.78	97.37	84.38	87.84	100	44.44	94.08	88.77
ResNet50	100	88.52	97.78	98.68	85.94	94.59	100	77.78	94.93	92.91
ConvNeXt	100	98.36	97.78	96.05	96.88	79.73	100	55.56	94.50	90.54

The confusion matrix (Figure 7) shows that the model achieves high accuracy across most vehicle colors, with perfect prediction for white, silver and red. The main source of misclassification occurs between gray and black, which is expected given that their visual similarity depends on natural illumination variance.

When it comes to specific real-world scenarios, captured vehicle colors can be visually ambiguous (gray vs. black in low light, green vs. blue in shadows, silver vs. white in bright sunlight) and in such cases it may be useful to consider an additional second choice. As it may be observed in Table 4, excellent results were obtained on the Top-2 evaluation, with MobileNetV3 and ConvNeXt with the highest weighted accuracy, closely followed by ResNet50. Compared to the Top-1 setting, weighted accuracy improved by 4.22% for EfficientNetV2, 5.50% for MobileNetV3, 5.07% for ViT-B16, 4.44% for ResNet50, and 5.08% for ConvNeXt.

Table 5 presents the F1-Score per class and the overall average for each tested model. All architectures demonstrate strong performance across the majority of color categories, with average F1-scores exceeding 90%, confirming the effectiveness of the chosen training strategy and data preprocessing.

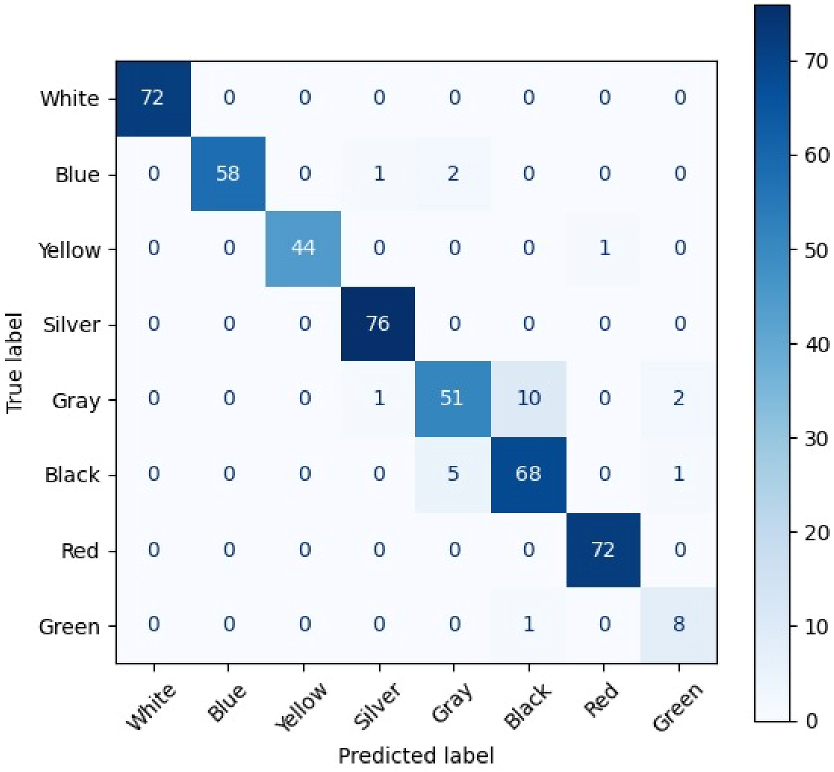


Figure 7. Color classification confusion matrix.

Table 4. Color classification accuracy (Top-2).

Model	White	Blue	Yellow	Silver	Gray	Black	Red	Green	Weighted Acc	Avg. Acc
EfficientNetV2	100	96.72	100	100	100	98.65	100	88.89	99.15	98.03
MobileNetV3	100	98.36	100	100	100	100	100	88.89	99.58	98.41
ViT-B16	100	100	97.78	100	98.44	100	100	44.44	99.15	96.75
ResNet50	100	98.36	100	100	98.44	100	100	88.89	99.37	98.21
ConvNeXt	100	100	100	100	100	98.65	100	88.89	99.58	98.44

Table 5. Color classification F1-Score.

Model	White	Blue	Yellow	Silver	Gray	Black	Red	Green	Avg. F1-Score
EfficientNetV2	100	97.48	98.88	98.70	83.61	88.89	99.31	80	93.36
MobileNetV3	97.14	97.52	98.88	92.11	86.52	90.78	99.31	82.35	93.08
ViT-B16	100	97.56	98.88	97.37	84.38	85.53	99.31	61.54	90.57
ResNet50	100	93.91	98.88	94.94	88	91.50	99.31	82.35	93.61
ConvNeXt	100	95.24	98.88	95.42	87.94	88.72	99.31	66.67	91.52

Among the models, ResNet50 and EfficientNetV2 achieved the highest overall scores, showing both stability and balanced precision–recall trade-offs across all color classes. MobileNetV3 follows closely, offering a competitive accuracy while maintaining computational efficiency, which makes it suitable for deployment on resource-constrained platforms. ConvNeXt and ViT-B16 exhibit slightly lower mean performance, mainly due to reduced accuracy in minority or unbalanced classes, particularly “Green”, which contains fewer samples.

Across all models, the “Red”, “White”, and “Yellow” categories achieved near-perfect scores, indicating strong chromatic distinctiveness and well-represented examples in the dataset. In contrast, classes such as “Gray” and “Green” remain more challenging due to subtle color differences and limited training data.

Overall, the results confirm that CNN-based architectures, particularly ResNet50 and EfficientNetV2, provide the most consistent and reliable performance for the vehicle color recognition task.

Table 6 outlines the Cohen's Kappa coefficients obtained for each evaluated model. All architectures achieved values above 0.93, indicating a very high level of agreement between the predicted and true color labels beyond chance. This confirms the overall consistency and reliability of the classification results across models. The highest Kappa value was obtained by both EfficientNetV2 and ResNet50, reinforcing their superior stability and balanced predictions observed in previous performance metrics. MobileNetV3, ConvNeXt, and ViT-B16 also achieved similarly strong agreement levels, reflecting that even with architectural differences, all networks generalize well across the dataset.

Table 6. Cohen's Kappa coefficient.

Model	Cohen's Kappa
EfficientNetV2	0.94
MobileNetV3	0.93
ViT-B16	0.93
ResNet50	0.94
ConvNeXt	0.93

Overall, the high Kappa coefficients validate that the models' outputs are not only accurate but also statistically consistent, demonstrating robustness in the vehicle color recognition task.

In addition, it is important to note that the selected color categories reflect real traffic trends, as colors such as white, silver, black, gray dominate the vehicle population, while colors like green and yellow appear far less frequently.

As mentioned in the previous subsection, regarding the make-model classification task, we conducted a hyperparameter tuning experiment to evaluate how different HOG configurations and SVM kernel settings influence the performance of the make-model classifier. For each tested configuration, we trained an SVM using the RBF kernel while varying the regularization factor C. The results of these experiments are illustrated in Figure 8, which presents the variation in accuracy as a function of the HOG spatial configuration and SVM hyperparameters.

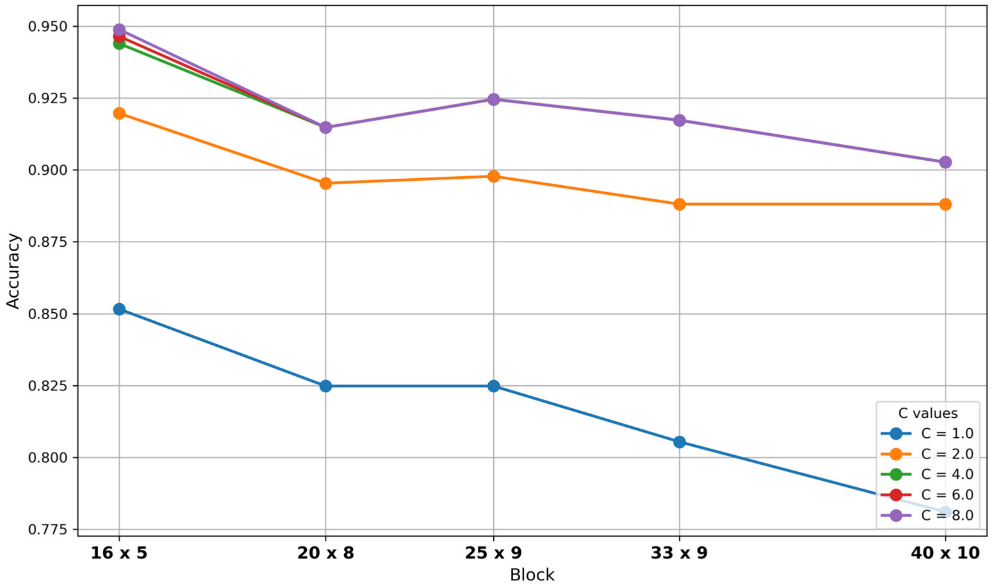


Figure 8. Make-model classification accuracy for different hyperparameter configurations.

The graph shows that the classifier performance is sensitive to both the spatial granularity of the extracted HOG features and the SVM kernel parameters. Higher accuracy values are obtained when using finer block structures and larger regularization values, demonstrating the benefit of representing more detailed gradient information. The best performance was achieved using the RBF kernel with γ = 1.0, C = 8, and a (16, 5) block dimension.

The performance of the make-model recognition module is summarized in Table 7.

Table 7. Make-model classification results.

Class	Precision	Recall	F1-Score
BMW 5 Series 2009–2013	100	88.89	94.12
BMW X3 2010–2013	90.91	100	95.24
Dacia Logan, MCV, Sandero 2016–2020	95.74	97.83	96.77
Dacia Duster 2013–2021	100	96.43	98.18
Dacia Duster 2024	92.31	92.31	92.31
Dacia Logan, MCV 2004–2012	89.66	96.30	92.86
Dacia Logan, Sandero 2020–2024	96.88	96.88	96.88
Dacia Spring 2021	100	66.67	80
Ford Focus 2004–2008	90	90	90
Renault Megane 2003–2009	76.92	100	86.96
Renault Megane 2016–2020	100	90	94.74
Renault Clio 2013–2019	100	90	94.74
Skoda Octavia 2004–2008	100	100	100
Skoda Octavia 2019–2024	100	100	100
Toyota Auris 2013–2018	100	75	85.71
Volkswagen Golf 2000–2003	96.97	100	98.46
Volkswagen Golf 2008–2012	88.24	100	93.75
Volkswagen Passat 2000–2005	93.33	82.35	87.50
Volkswagen Passat 2005–2010	93.75	97.83	95.74
Volkswagen Passat 2010–2014	100	85.71	92.31
Volkswagen Passat 2014–2019	91.67	91.67	91.67
Volkswagen Tiguan 2008–2011	100	95	97.44
Volkswagen Touran 2003–2006	93.75	100	96.77
Volkswagen Up 2012–2023	100	100	100
Overall	95.42	93.04	93.84

The system achieved an overall accuracy of 94.89%. In general, most classes reached high precision and recall with some achieving perfect scores, such as "Skoda Octavia 2004–2008", "Skoda Octavia 2019–2024" and "Volkswagen Up 2012–2023". Unfortunately, performance dropped for a few classes. For instance, "Renault Megane 2003–2009" obtained relatively low precision, while "Dacia Spring 2021" reached only 66.67% recall. The majority of classes show balanced precision and recall values above 90%, confirming that the model not only predicts correctly but also detects most instances of each class, even when inter-class similarity exists. A few classes present larger differences between precision and recall, such as "BMW 5 Series 2009–2013" and "Toyota Auris 2013–2018", indicating that when the model predicts these classes, it is usually correct, but it sometimes fails to detect them. Another example is "Dacia Spring 2021", with 100% precision and the lowest recall, meaning no false positives occurred, but some instances were missed. This indicates that when vehicle exhibit distinctive frontal characteristics, the model can reliably separate them for other classes. Some make-model-generation combinations share very similar frontal design features, which increases the likelihood of misclassification. However, in real-world deployments, the system must be able to handle a wide variety of makes, models and generations, meaning that having as many classes as possible is crucial.

5. Conclusions

This work shows that a segmentation-guided Automatic Vehicle Recognition system can deliver strong, balanced performance in both VCR and VMMR within a single, region-tailored framework. The system uses a pre-trained YOLOv8 detector for localizing vehicles in the images, and then a semantic-segmentation node to obtain fine-grained, part-aware masks.

On VCR, all backbones achieve high Top-1 (weighted ≈ 94–95%), with EfficientNetV2 and ResNet-50 leading weighted accuracy, and MobileNetV3 close behind; Top-2 pushes every model to ≈ 99%, confirming that many real-world ambiguities (gray ↔ black, blue ↔ green, silver ↔ white) are resolved once a second choice is allowed—an effect consistent with findings that color can be visually ambiguous in operational imagery, and that relaxing decision strictness is helpful on harder datasets. The ViT-B/16 low accuracy on green reflects a common pattern: minority classes remain fragile unless the training data are explicitly rebalanced or augmented. Our ResNet-50 color performance aligns with on-road pipelines reporting ≈ 94% color accuracy ([4]—vehicle color top-1 accuracy 94.2%), while mask-guided color extraction echoes the benefit of part/region segmentation for cleaner chromatic cues [21].

For VMMR, an overall accuracy of 94.89% with macro precision/recall/F1 = 0.9542/0.9304/0.9384 is competitive with standard fine-tuning baselines on curated datasets and consistent with typical intra-domain results (as summarized in [12], and dataset works in [13,14]). Models perform best when distinctive front-end details (grille, headlamps, and bumper) are visible, but they struggle to tell apart look-alike generations. This points to the value of nudging the model to focus on the most informative parts. It also helps to use training practices that keep performance stable when data are noisy or unbalanced.

Methodologically, we add a segmentation step that: keeps only the body panels for color recognition, and highlights front-end details for make-model recognition. This cuts background clutter and feeds each task the most relevant pixels. Still, some limits remain: rare colors stay tricky, footage from different sources does not always match, and a few classes are much less common than others.

The YOLOv8 detector for vehicle localization and DeepLabv3 for semantic segmentation, while not representing the latest state-of-the-art techniques, were integral to the dataset generation process and remain highly optimized and efficient for deployment. We project that utilizing more robust detection methods would yield subtle improvement in the VMMR task, given the high accuracy already achieved and the targeted nature of the Region of Interest (ROI) extraction. A greater potential gain is acknowledged for the VCR task, where state-of-the-art segmentation models could provide cleaner, more fine-grained pixel masks for color sampling, potentially subtly increasing overall accuracy.

Going forward, we will: broaden coverage of tough conditions like nighttime and heavy rain, rebalance rare colors and models with targeted data augmentation, and refine and expand our datasets so they keep up with the changing vehicle market.

Author Contributions: Conceptualization, A.I., M.-G.B., D.-T.H. and F.R.; methodology, A.I., M.-G.B., D.-T.H. and F.R.; investigation, A.I. and M.-G.B.; resources, A.I., M.-G.B., D.-T.H., F.R. and C.G.; writing—review and editing, F.R., C.G., A.I., M.-G.B., D.-T.H. and M.E.; supervision, F.R., A.I. and M.E.; project administration, F.R.; funding acquisition, F.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CCCDI—UEFISCDI, project number PN-IV-P6-6.3-SOL-2024-0060, within PNCDI IV.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to contractual obligations associated with the project in which the dataset was developed, the dataset is unable to be made publicly available. Instead, the data can be provided upon request, conditional on the signing of a non-disclosure agreement (NDA).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Neagu, A.-C.; Ciubotaru, B.-I. Use cases of artificial intelligence in military systems. In Proceedings of the 20th International Scientific Conference "Strategies XXI Technologies—Military Applications, Simulation And Resources", Bucharest, Romania, 28 March 2024; "Carol I" National Defence University Publishing House: Bucharest, Romania, 2024; Volume 20, pp. 151–155.
2. Sadiq, S.; Sultan, K.; Sheraz, M.; Chuah, T.C.; Hashmi, M.U. Towards Efficient Vehicle Recognition: A Unified System for VMMR, ANPR, and Color Classification. *Comput. Mater. Contin.* **2025**, *85*, 3945–3963. [CrossRef]
3. Munoz, A.; Thomas, N.; Vapsi, A.; Borrajo, D. Veri-Car: Towards open-world vehicle information retrieval. *Neural Comput. Appl.* **2025**, *37*, 15183–15221. [CrossRef]
4. Kim, J. Deep learning-based vehicle type and color classification to support safe autonomous driving. *Appl. Sci.* **2024**, *14*, 1600. [CrossRef]
5. Varghese, R.; Sambath, M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. [CrossRef]
6. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587. [CrossRef]
7. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2024; Volume 1, pp. 886–893. [CrossRef]
8. Tan, M.; Le, Q.V. EfficientNetV2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning (PMLR), Virtual Event, 18–24 July 2021; Volume 139, pp. 10096–10106.
9. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2020; pp. 1314–1324. [CrossRef]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF, International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 12 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778. [CrossRef]
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2021**, arXiv:2010.11929. [CrossRef]
12. Gayen, S.; Maity, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Two decades of vehicle make and model recognition–survey, challenges and future directions. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 101885. [CrossRef]
13. Tafazzoli, F.; Frigui, H.; Nishiyama, K. A large and diverse dataset for improved vehicle make and model recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8. [CrossRef]
14. Buzzelli, M.; Segantin, L. Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments, and results. *Sensors* **2021**, *21*, 596. [CrossRef]
15. Lyu, Y.; Schiopu, I.; Cornelis, B.; Munteanu, A. Framework for vehicle make and model recognition—A new large-scale dataset and an efficient two-branch–two-stage deep learning architecture. *Sensors* **2022**, *22*, 8439. [CrossRef] [PubMed]
16. Komolovaite, D.; Krisciunas, A.; Lagzdinyte-Budnike, I.; Budnikas, A.; Rentelis, D. Vehicle make detection using the transfer learning approach. *Elektron. Elektrotechnika* **2022**, *28*, 55–64. [CrossRef]
17. Zhang, C.; Li, Q.; Liu, C.; Zhang, Y.; Zhao, D.; Ji, C.; Wang, J. A Fine-Grained Car Recognition Method Based on a Lightweight Attention Network and Regularized FineTuning. *Electronics* **2025**, *14*, 211. [CrossRef]
18. Manzoor, M.A.; Morgan, Y.; Bais, A. Real-time vehicle make and model recognition system. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 611–629. [CrossRef]
19. Lv, C.; Kumari, S.; Singh, P.; Wang, H.; Kumar, S.; Liu, W.; Madaan, V.; Agrawal, P. Vehicle Detection and Classification Using an Ensemble of EfficientDet and YOLOv8. *PeerJ Comput. Sci.* **2024**, *10*, e2233. [CrossRef] [PubMed]
20. Hu, M.; Wu, Y.; Fan, J.; Jing, B. Joint semantic intelligent detection of vehicle color under rainy conditions. *Mathematics* **2022**, *10*, 3512. [CrossRef]

21. Stavrothanasopoulos, K.; Gkountakos, K.; Ioannidis, K.; Tsikrika, T.; Vrochidis, S.; Kompatsiaris, I. Vehicle Color Identification Framework using Pixel-level Color Estimation from Segmentation Masks of Car Parts. In Proceedings of the 5th International Conference on Image Processing Applications and Systems (IPAS), Genova, Italy, 5–7 December 2022; IEEE: Piscataway, NJ, USA, 2023; pp. 1–7. [CrossRef]
22. Lima, G.E.; Laroca, R.; Santos, E.; Nascimento, E.; Menotti, D. Toward enhancing vehicle color recognition in adverse conditions: A dataset and benchmark. In Proceedings of the 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Manaus, Brazil, 30 September–3 October 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. [CrossRef]
23. Ayub, A.; Kim, H. GAN-based data augmentation with vehicle color changes to train a vehicle detection CNN. *Electronics* **2024**, *13*, 1231. [CrossRef]
24. Semiromizadeh, N.; Manzari, O.N.; Shokouhi, S.B.; Mirzakuchaki, S. Enhancing Vehicle Make and Model Recognition with 3D Attention Modules. In Proceedings of the 14th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 19–20 November 2024; IEEE: Piscataway, NJ, USA, 2025; pp. 87–92. [CrossRef]
25. Nafzi, M.; Brauckmann, M.; Glasmachers, T. Vehicle shape and color classification using convolutional neural network. *arXiv* **2019**, arXiv:1905.08612. [CrossRef]
26. Liu, D. Progressive multi-task anti-noise learning and distilling frameworks for fine-grained vehicle recognition. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 10667–10678. [CrossRef]
27. Wolf, S.; Loran, D.; Beyerer, J. Knowledge-distillation-based label smoothing for fine-grained open-set vehicle recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV Workshop 2024, Waikoloa, HI, USA, 1–6 January 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 330–340.
28. Sielemann, A.; Wolf, S.; Roschani, M.; Ziehn, J.; Beyerer, J. Synset boulevard: A synthetic image dataset for VMMR. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 9146–9153. [CrossRef]
29. Berwo, M.A.; Khan, A.; Fang, Y.; Fahim, H.; Javaid, S.; Mahmood, J.; Abideen, Z.U.; M.S., S. Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey. *Sensors* **2023**, *23*, 4832. [CrossRef]
30. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1492–1500. [CrossRef]
31. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 11966–11976. [CrossRef]
32. Liu, X.; Liu, W.; Zheng, J.; Yan, C.; Mei, T. Beyond the parts: Learning multi-view cross-part correlation for vehicle re-identification. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 907–915. [CrossRef]
33. Todi, A.; Narula, N.; Sharma, M.; Gupta, U. ConvNext: A Contemporary Architecture for Convolutional Neural Networks for Image Classification. In Proceedings of the 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, 8–9 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. [CrossRef]
34. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255. [CrossRef]
35. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.