AI

MDPI

Article

Artificial Intelligence for Text-Based Vehicle Search, Recognition, and Continuous Localization in Traffic Videos

Karen Panetta [1], Landry Kezebou [1,*], Victor Oludare [1], James Intriligator [1] and Sos Agaian [2]

[1] Department of Electrical & Computer Engineering, School of Engineering, Tufts University, Medford, MA 02155, USA; karen@eecs.tufts.edu (K.P.); victor.oludare@tufts.edu (V.O.); james.intriligator@tufts.edu (J.I.)
[2] Department of Computer Science, School of Engineering, City University of New York (CUNY), New York, NY 10031, USA; sos.agaian@csi.cuny.edu
* Correspondence: landry.kezebou@tufts.edu

Abstract: The concept of searching and localizing vehicles from live traffic videos based on descriptive textual input has yet to be explored in the scholarly literature. Endowing Intelligent Transportation Systems (ITS) with such a capability could help solve crimes on roadways. One major impediment to the advancement of fine-grain vehicle recognition models is the lack of video testbench datasets with annotated ground truth data. Additionally, to the best of our knowledge, no metrics currently exist for evaluating the robustness and performance efficiency of a vehicle recognition model on live videos and even less so for vehicle search and localization models. In this paper, we address these challenges by proposing **V-Localize**, a novel artificial intelligence framework for vehicle search and continuous localization captured from live traffic videos based on input textual descriptions. An efficient **hashgraph** algorithm is introduced to compute valid target information from textual input. This work further introduces two novel datasets to advance AI research in these challenging areas. These datasets include (**a**) the most diverse and large-scale Vehicle Color Recognition (**VCoR**) dataset with **15 color** classes—twice as many as the number of color classes in the largest existing such dataset—to facilitate finer-grain recognition with color information; and (**b**) a Vehicle Recognition in Video (**VRiV**) dataset, a first of its kind video testbench dataset for evaluating the performance of vehicle recognition models in live videos rather than still image data. The **VRiV** dataset will open new avenues for AI researchers to investigate innovative approaches that were previously intractable due to the lack of annotated traffic vehicle recognition video testbench dataset. Finally, to address the gap in the field, **five novel metrics** are introduced in this paper for adequately accessing the performance of vehicle recognition models in live videos. Ultimately, the proposed metrics could also prove intuitively effective at quantitative model evaluation in other video recognition applications. T One major advantage of the proposed vehicle search and continuous localization framework is that it could be integrated in ITS software solution to aid law enforcement, especially in critical cases such as of amber alerts or hit-and-run incidents.

Keywords: text-based search; vehicle recognition; vehicle make model year and color recognition; object detection; object recognition in the dark; vehicle recognition in videos; VCoR dataset; VRiV dataset; evaluation metrics

check for updates

Citation: Panetta, K.; Kezebou, L.; Oludare, V.; Intriligator, J.; Agaian, S. Artificial Intelligence for Text-Based Vehicle Search, Recognition, and Continuous Localization in Traffic Videos. *AI* **2021**, *2*, 684–704. https://doi.org/10.3390/ai2040041

Academic Editor: Gianni D'Angelo

Received: 5 November 2021
Accepted: 2 December 2021
Published: 6 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The ability to recognize a vehicle by specific features such as its make, its model, or year of manufacture, is becoming a much-needed solution to provide advanced security features for intelligent transportation and traffic surveillance systems. Plate number recognition is no longer sufficient because plates can be tampered with, swapped, altered, or deliberately occluded as shown in Figure 1. It would be more efficient and more accurate to locate a "Silver Toyota Prius third generation (2010–2015) with plate number 9GB217" than to locate any vehicle bearing the plate number 9GB217, especially in cases of Amber alerts,

hit and run incidents, or other forms of criminal activities [1,2]. Additionally, it is more common for individuals to remember a vehicle make/model (perhaps along with a partial license plate) than to remember an exact license. Intuitively, text-based searches can be more synergistic with the manner most incidents are reported by eyewitness accounts.

Figure 1. Examples of plate numbers exhibiting potential challenges for plate number identification systems.

The continued growth in traffic surveillance camera networks has imposed severe cognitive load pressures on traffic control agents who are overwhelmed by the vast amount of live surveillance footage that requires real-time monitoring and intervention where necessary. As such, there is increased potential that numerous criminal activities and incidents can go unnoticed or unsolved for longer periods of time due to a lack of optimization in information search and subsequent resource deployment. Furthermore, Departments of Transportation struggle with responding to requests for incident data from insurance companies because the time and manual resources it takes to search among vast quantities of information. In fact, in some traffic centers rules are put in place requiring the deletion of any data older than, for example, 30 days. These rules are, at least in part, established to help avoid responding to many data requests. To help assuage these issues, recently several assistive technologies have been proposed for applications such as incident detection [1,3–8].

Interest in vehicle model and make recognition (VMMR) has been on the rise for various reasons including, advanced security, targeted advertising, faster surveillance, and customer segmentation. The release of large-scale datasets such as VMMR [9], Stanford [10], and CompCars [11] have inspired the development of numerous computer vision models for vehicle recognition tasks [1,6,12–14]. However, these existing methods work best on high quality still input image data in which distinctive features such as the manufacturer logo, the bumper, the headlights, taillights, and chassis are visible. Inevitably, these recognition models' performances decline if not provided with high quality input data, thereby impairing their ability to perform real-time recognition in live-traffic video feeds. This is primarily because of data quality and quantity: cropped patches of detected vehicles from a video stream are very small and blurry, meaning distinctive features are much harder to recognize. Hence, it is important to incorporate such challenges in the training process for improved performance, since it is a gating issue preventing vehicle recognition algorithms from working well on live video feeds.

Additionally, searching and localizing a vehicle based on its textural description such as a sentence, a paragraph, or a report could significantly boost security systems' capability for solving crimes much faster. Consider for example, the case of an amber alert where an eyewitness reports an incident. Such a report may or may not contain the accurate plate number description, but other vital information such as vehicle color, make, model, and the location the vehicle was last seen contained in the report could prove crucial in the timely solving and safe recovery of the victim. It is intractable to manually locate such a vehicle from surveillance cameras. It is even more computationally expensive to process the number plate for all vehicles present in a video when searching for the target vehicle. A more efficient way of quickly localizing such a target vehicle in video would be

to perform a content-based search where the model searches for a specific target based on a textural description. Such text-based search algorithms for vehicle recognition have yet to be explored.

In this paper, a novel artificial intelligence framework for text-based vehicle search, recognition, and continued localization in traffic videos is proposed to address the above-mentioned shortcomings of existing models. Figure 2 provides a pictorial view of the proposed framework for text-based vehicle search and continuous localization, in comparison with the existing frameworks used in modern vehicle recognition models.

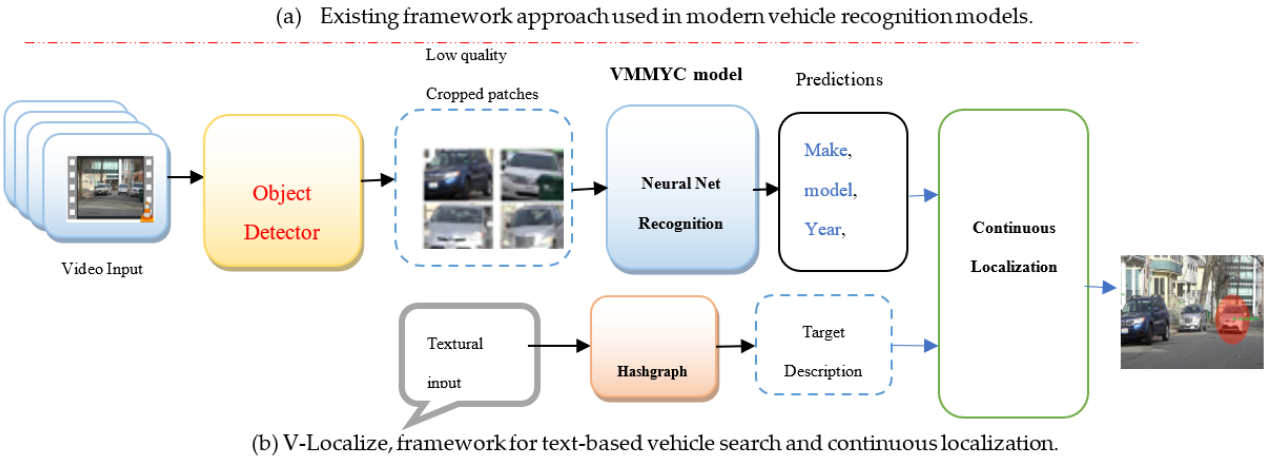

Figure 2. Comparison between (**a**) the framework for modern vehicle recognition models and (**b**) the proposed novel text-based vehicle search and continuous localization framework (V-Localize). The V-Localize framework can operate on videos more efficiently based on its training loop and achieve finer-grain recognition with a color component added. Lastly, V-Localize can search for targets based on input textural description.

The research objectives of this paper can be summarized into the following contributions:

(a) a hashgraph algorithm is introduced for automatically computing the search query from the input text at high speed;

(b) a large scale and diverse vehicle color recognition dataset (VCoR) with 15 color classes and more than 10k images samples is proposed to facilitate training finer-grain vehicle recognition including a color description component. The creation of the VCoR dataset was motivated by the fact that the largest existing vehicle color dataset only has eight color classes, and as such will limit the localization capability (specificity) of the proposed framework.

(c) V-Localize, a deep learning-based framework for text-based vehicle search and continuous localization, is proposed. A finer-grain VMMYC (Vehicle Make Model Year and Color) recognition model is trained for efficiently matching valid search targets. The training procedure is designed to account for real-world traffic scenarios and ultimately address limitations of the existing models, which are best suited for still image inputs.

(d) A first of its kind VRiV (Vehicle Recognition in Video) testbench dataset for benchmarking the performance of automatic vehicle recognition models in video feeds is introduced. This dataset is aimed at addressing the lack of such testbench video datasets, which are required to drive research for vehicle recognition models more suited for live video input.

(e) Lastly, because there currently exist no metrics for adequately accessing the robustness and efficiency of recognition and continuous localization models, novel metrics are intuitively proposed to address this shortcoming. These metrics establish a foundation for qualitatively benchmarking and comparing future models on recognition tasks on live video.

The remainder of the paper is organized as follows: Section 2 explores the related literature; Section 3 presents the details of the proposed framework and its submodules; while in Sections 4–6 the datasets and evaluation metrics; simulation and performance results; conclusion, limitation, and future work, are presented respectively.

2. Related Work

As application opportunities for vehicle recognition systems continue to expand, practical methods for vehicle make and model recognition (VMMR) have evolved. Earlier classical methods used more traditional approaches for detecting vehicles in scenes and leaned towards the use of feature descriptors such as SIFT [15], SURF [16,17], and HOG [18] to retrieve relevant features, and using traditional machine learning techniques such as SVM [19,20], K-Means clustering [21,22], and Random Forest [23]. More recent methods have favored leveraging deep convolutional neural network methods such as MobileNet [24], ResNet [25], SqueezeNet [26], EfficientNet [27], and VGG [28], for training the model to automatically represent images in a hyperspace that is large and sufficiently complex to distinctively separate vehicles of different make and designs. These later techniques usually include a localization step using state-of-the-art object detectors such as YOLO [29], RCNN [25], Fast-RCNN [24], and SSD [30].

Kazemi et al. [31] proposed a vehicle recognition algorithm using the curvelet transform and a SVM classifier. The authors exploit unique properties of the curvelet transform such as time-frequency localization, degree of directionality and anisotropy, and scale and orientation of gradients, to extract relevant descriptive features from vehicle images, which are then used to train an SVM classifier to distinguish between classes. However, their study was only conducted on a limited dataset of 300 images containing only five vehicle classes; hence, the approach is prone to overfitting. Pearce [13] et al. championed using the Harris corner feature extractor and experimented with "make and model" classification using Naive Bayes and KNN classifiers. This study was also conducted on a limited set of data, and also presented numerous limitations in terms of input type, task complexity, training approach, and inherent limitations in the classifier chosen.

Manzoor et al. [32] proposed a vehicle make and model classification method using a bag of SIFT features. The approach consists of segregating regions of interest from which visual features are extracted using SIFT and clustered using a bag of words. SVM is used to train on extracted features to enable proper classification. Similar to this method is the bag of SURF (BoSUFT) method introduced by Siddiqui et al. [6] for vehicle make and model recognition. These approaches are limited by the feature extractor technique employed and the resultant accuracies have been outpaced by more recent state-of-the-art performance techniques. Boonsim and Prakoonwit [12] explored performing recognition by using a combination of taillights shape features, license plates, rearview, and other salient geographical features. These features are then concurrently passed to multiple machine learning classifiers including SVM, decision trees, and K-nearest neighbors, from which target verification is performed by majority vote. A recent method for vehicle make and model recognition introduced by Wang et al. [33] suggests using the multiple feature subspace hypothesis based on sparse constraints, and transfer learning. Deep Belief Network (BDN) is used in combination with multiple Restricted Boltzmann Machines (RBM) to constitute the multiple subspace feature extractor. This method presents several limitations in terms of time complexity, ambiguity in determining the number of subspaces required for optimal performance, hence, making it unsuitable for practical realtime applications.

More advanced deep neural network methods have also helped push the boundaries of vehicle recognition, especially as more datasets became available. Fang et al. [34] proposed using coarse-to-fine convolutional neural network architecture for achieving fine-grained vehicle model and make recognition. Fang et al. identifying discriminative regions from which local and global features and cues are extracted via a hierarchical styled neural network. The Network model learns to extract local features from distinctive

regions, and global features from the whole vehicle, achieving state-of-the-art accuracy on 281 vehicle make and model classes.

Ma et al. [35] proposed an AI-based visual attention model for vehicle make and model recognition. Ma et al. proposed an expansion of classical CNN architecture for VMMR systems by introducing a Recurrent Attention Unit (RAU), a modular unit which is applied at multiple layers of the vanilla CNN to enhance learning objective of accurately discriminating local regions of vehicles at multiple scales. The authors experiment with ResNet101-RAU, a modified version of ResNet101 with an added RAU, on the popular Stanford and CompCars dataset. ResNet-RAU also achieved state-of-the-art-accuracy. Another method to mention is the vehicle recognition with Residual SqueezeNet by Lee et al. [14]. Lee et al. argued that introducing residual connection in the original SqueezeNet helps extract more distinctive features for vehicle make and model recognition. Lee et al. further applied principal component analysis (PCA) [36] to reduce feature dimensionality space, and k-means clustering was used for final classification.

Although the later methods exhibited better performance than classical methods, they are still faced with numerous challenges and limitations. First, even modern recognition models fail to consider valuable details such as vehicle color which would otherwise make the model more robust and more adequate for practical application in security systems. Second, the accuracy of existing models is often reported on still images (often of high quality), as such, it is difficult to replicate reported accuracies on live video footage which simulates real-world conditions much better than the ideal still images chosen as a validation set.

In this paper, the aforementioned shortcomings of the existing vehicle model and make recognition systems are addressed by developing a more robust model which is more suitable for real-world applications such as live traffic videos. The approach considers recognizing vehicles not only by their make, model, and year but also includes a color component making the recognition more specific. It is much more specific and helpful for security reasons to locate a "Grey Toyota Prius 2014" on a highway road, than it is to locate any "Toyota Prius" or any "Toyota Prius 2014". The color inclusion provides a narrower scope, which in turn can minimize the time complexity required to locate a vehicle in a scene, enabling a more responsive feature for traffic intelligence systems. Several neural network architectures are investigated including ResNet [37], ResNet-CBAM (ResNet with CBAM- Convolution Block Attention Module) [38,39], MobileNetV3 [40], and RepVGG [41] architectures. Empirically, RepVGG architecture shows the greatest advantage to solving the challenges. Hence, RepVGG is adopted as the backbone in the proposed framework. Multiclass classification is achieved by modifying the classification layer and corresponding optimization function. The preprocessed training data are used to simulate real-world scenarios, which are robust and adaptable for recognition in live-video feed. Finally, the concept of text-based vehicle search, and continuous localization based on input textural description, is introduced demonstrating the tremendous potential for real-world applications such as amber alerts on highway roads.

3. Materials and Methodology

This section provides a more detailed description of the system flow architecture. Dataset preparation and training pipeline is presented. The Hashgraph algorithm is also presented, and its space and time complexity analyzed. Furthermore, the novel VCoR dataset proposed in this paper is discussed in detail, and its impacts on training the finer-grain recognition model emphasized as shown in Figure 3.

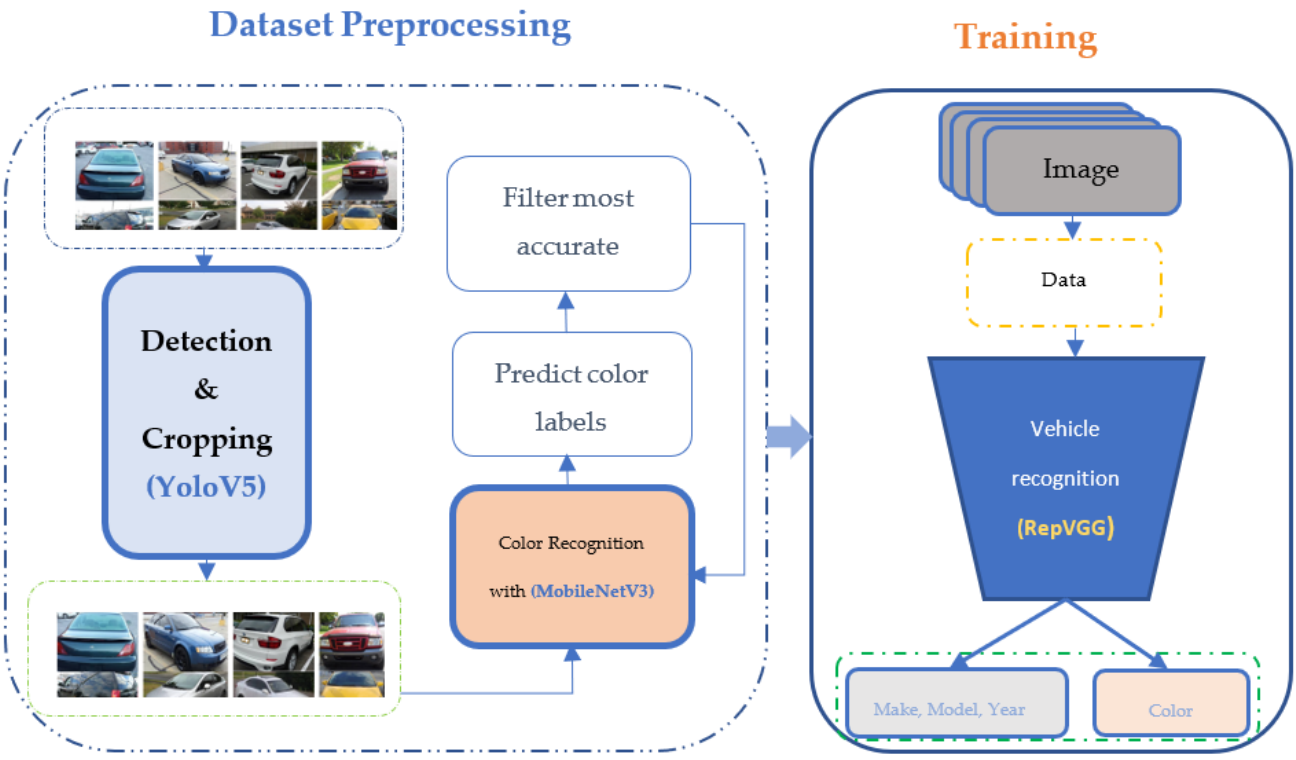

Figure 3. Training pipeline for the VMMYC recognition model. The module on the left represents the data preparation step, while that on the right exhibits the training loop.

3.1. System Flow

The proposed V-Localize, the text-based vehicle search, recognition, and continuous localization framework, is an efficient integration of four modules into a customizable pipeline to facilitate high speed operation. The pipeline includes: (a) an object detection module for detecting vehicles patches from the video feed; (b) a RepVGG-based multi-class-multi-label classification model for finer-grain vehicle recognition by make, design, year of manufacture, and color; (c) a Hashgraph module for managing search requests and efficiently computing specific target descriptions such as make, model, year, and color, by processing the textural input that might be found in a police report, SMS, or an eyewitness report; and (d) a continuous localization module to locate and validate matching targets and keep track of them.

Figure 2 shows the system flow and architectural diagram of the proposed Text-Based Vehicle Search, Localization, and Tracking Framework. The YoloV5 object detector is selected for its high speed and high accuracy. Furthermore, the object detector is fine-tuned to only process objects of interest (including cars, buses, trucks, motorcycles, etc.), which are relevant to the tasks at hand. The detection is run in real-time on live video feed to avoid computational overhead due to recurrent reinitialization. The vehicle recognition model is triggered preemptively by the hashgraph algorithm when raw textural input is detected, and at least one valid candidate target is computed. The vehicle recognition is trained on the VMMR dataset augmented with data samples, which simulate cropped patches of vehicles from video feeds. Detected vehicles are cropped into patches which are resized to 224 $\times$ 224 and then processed by the vehicle recognition model (in batches to speed up operation). Predicted target labels are thereafter compared with the computed search target to localize the target of interest. To further minimize computational overhead, the recognition module could be run every five frames instead of every frame without significant performance loss.

3.2. Training Pipeline for the Finer-Grain Vehicle Recognition Model

The proposed **VMMYC** (Vehicle Make Model Year and Color) model is trained on the VMMR vehicle dataset [9] available for download here: https://github.com/faezetta/VMMRdb (accessed on 20 June 2021). The VMMR dataset is a large-scale vehicle dataset, which contains 291,752 vehicle images from 9170 distinct vehicle make and models. However, many of the classes in the VMMR dataset contain less than 10 images which is imbalanced and unsuitable for accurate training. Furthermore, the dataset covers vehicles make and designs manufactured between 1950 and 2016. It is very unlikely to encounter a vehicle from the 1950s in today's traffic, unless it is a classic or antique car. To remedy the noise potentially added by these rare cases, a more balanced dataset is segregated from the large pool by first, eliminating all classes with fewer than 40 samples. Next, a filter is applied to only select classes of vehicles between 1995 and 2016.

Additionally, vehicles of the same make and model, which were manufactured within a few years of each other will share enormous feature similarities, especially in terms of their external structure, shape, and other design features. It is therefore pertinent, to increase the number of image samples per class for improved training and testing accuracy. We accomplished this by combining vehicles of same make and design by brackets of 3 to 4 years—which seemed sensible given inherent feature similarities as exemplified in Figure 4.

It is also crucial to observe that most of the image samples in the VMMR dataset have noisy backgrounds often capturing, in great detail, other vehicles. Using such raw data could very much prevent the model from focusing on the vehicle of interest and learning the most important features and achieving high recognition accuracy. To address this issue, object detection is performed first on image samples and patches containing vehicles are cropped—eliminating most of the background noise as shown in Figure 5.

In a live-video feed, detected vehicles often appear very small and of low visual quality when upscaled to fit the recognition model input (224×224 in this case). Therefore, it is important to introduce such diversity in the training data to enable the model to sustain the same performance on live-video feed. This motivates the necessity to downscale and upscale images in the dataset to introduce such low visual quality input data as shown in Figure 6, thus augmenting the training.

Figure 4. Sample image data showing feature similarity between vehicles of the same make and model designs but manufactured within a few years of each other. These similarities inspired the grouping of vehicles manufactured within the space of 3 to 4 years, to increase the number of samples available per class. Note, the license plates have been purposely obscured to preserve privacy.

Figure 5. Sample images showing how noisy backgrounds are removed to improve training efficiency. The first row shows images with noisy background where other vehicles can be seen, oftentimes of different colors. The second shows the corresponding cropped images with background noise removed.

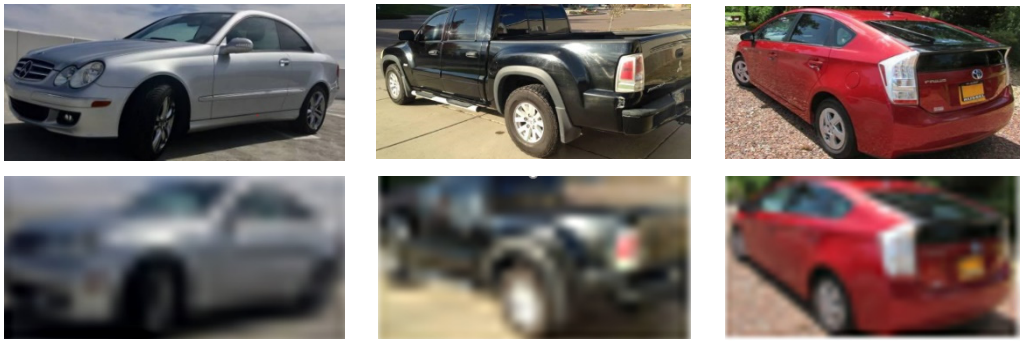

Figure 6. Examples of downscaling and upscaling images to introduce diversity in terms of image resolution and visual quality.

The final training set, without augmentation, contained approximately 242 k image samples for a curated 959 classes by make, model, and year. The dataset was then split into 70% training, 20% validation, and 10% testing.

The RepVGG backbone was empirically chosen for its efficiency and its light weight compared to other models such as ResNet, VGG, SqueezeNet, and MobileNetV3. Pretrained weights of the RepVGG architecture were utilized, but the final layers were modified to accommodate multilabel classification. The last three layers were also unfrozen to fine-tune the classifier during training. An *Adam* optimizer was empirically chosen via extensive computer simulations with a learning rate, weight decay, and momentum set as: $l_r = 0.01$, $m = 0.01$, and $w = 0.01$, respectively, for optimum performance. The learning rate schedule was experimentally chosen to be $l_r = l_r * \left(0.1^{\frac{current\ epoch}{35}}\right)$, and training was performed for 110 epochs.

To the best of our knowledge, the largest existing vehicle color recognition dataset is the one proposed by Chen et al. [42] with about 15k image samples. However, this dataset is limited to only eight color classes, which makes it less suitable for the finer-grain vehicle recognition based on textural description championed in this article. Very limited work has be done on vehicle color recognition [43–45] due to the paucity of datasets This lack motivated us to create a large scale and diverse vehicle color dataset (VCoR) containing 10k+ image samples and 15 color classes, which is almost twice as diverse as the largest existing dataset. The 15 color categories represent the most popular vehicle color models according to CarMax [46,47], including: white, black, grey, silver, red, blue, brown, green, beige, orange, gold, yellow, purple, pink, and tan.

Figures 7 and 8 show the sample distribution per color class and sample images from the proposed VCoR dataset, respectively.

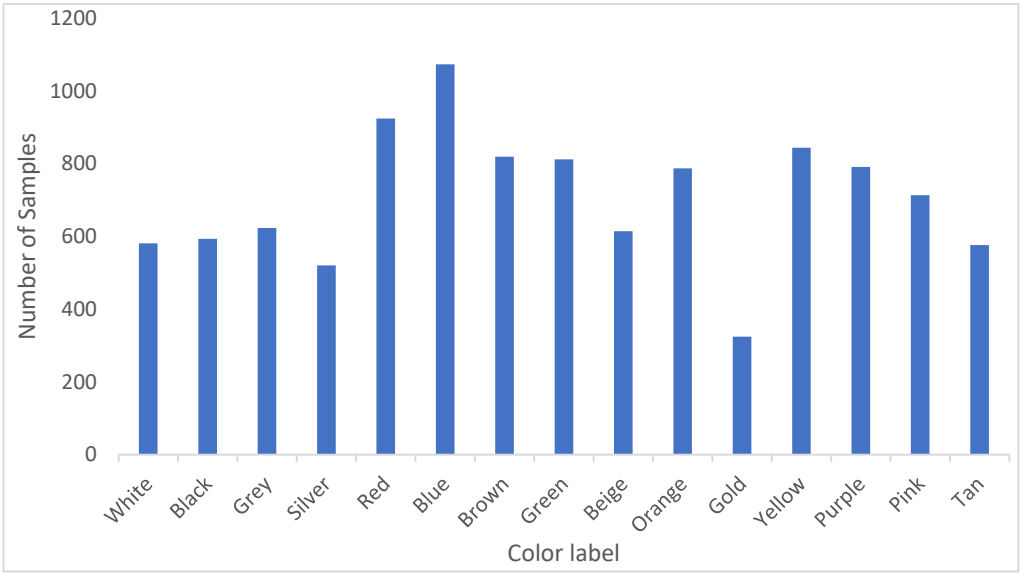

Figure 7. Image sample distribution across color classes in the proposed vehicle color recognition (VCoR) dataset.

To add color label annotations to the training data discussed above, an efficient vehicle color recognition model was trained using MobileNetV3 as shown in Figure 3. The training data were labelled in three passes. In the first pass, only the predicted color labels with confidence greater than 0.6 were saved as annotations for corresponding samples. The color-labelled data from the first pass were used to augment training data, and to fine-tune the trained color model in order to make a second prediction on samples with lower prediction confidence. The final round involved manually annotating samples with lower color predictions scores and cleaning up the data by adjusting the very few wrongly predicted labels.

Generated color labels enable multiclass multi-label training of the recognition pipeline, which predicts on one hand, the vehicle make, design, and year; and on the other hand, the vehicle color information. We experimentally found that treating the make, design, and year labels independently, significantly reduced the recognition model accuracy, because the probability for invalid target predictions such as Hyundai Civic 2012 (instead of Honda Civic), or Pontiac Matrix (instead of Toyota Matrix) became much higher.

3.3. Hashgraph for Converting Textural Input to Valid Search Target

The proposed hashgraph algorithm used in the V-Localize framework is based on a hybrid combination of hash maps and graph search algorithm. It processes the text input to automatically compute the target description in a format that is used to query the vehicle recognition model. First, a set of hash maps with unique class labels in terms of car make, model, year, and color, were generated. The hash maps were chosen for their computational efficiency in terms of time and space complexity. The space and time complexity for each hash map are defined as:

$$\begin{cases} \textit{time complexity} = O(1) \\ \textit{space complexity} = O(n) \end{cases} \quad (1)$$

where n represents the total number of labels for that particular class in the dataset.

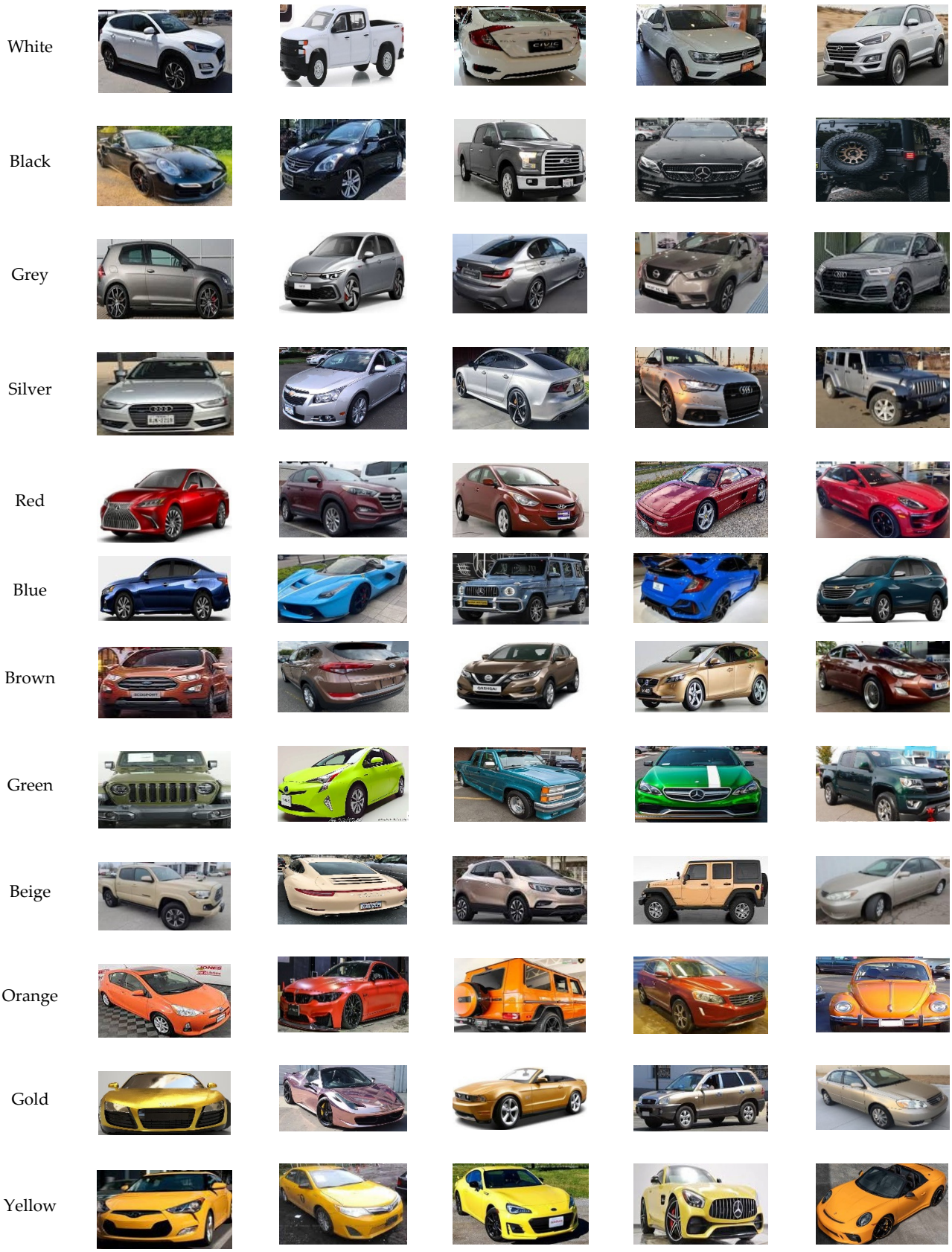

Figure 8. *Cont.*

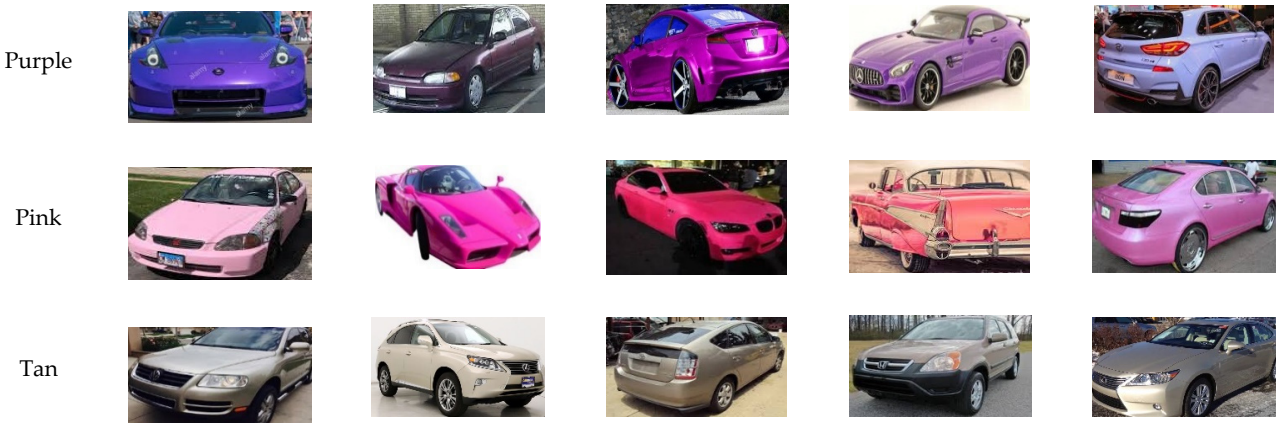

Figure 8. Sample images from the proposed vehicle color recognition model. The images are grouped by row based on color information, and each row shows multiple variants of the corresponding color label in the dataset.

The hash maps were used to extract cues and keywords from the input document which could be a sentence, a paragraph, or an entire report. Five hash maps, respectively for make, model, year, color and miscellaneous were maintained as shown in Figure 9. The miscellaneous dictionary contains other important cues. The total time and space complexity for the cues extraction from text is defined as:

$$\begin{cases} \textit{time complexity} = O(k) \\ \textit{space complexity} = O(n) \end{cases} \tag{2}$$

where k represents the number of words in the input document excluding punctuations.

The extracted cues/keywords were used to constitute a graph from which a breadth-first search algorithm was used to compute target description options. A validation step was used to eliminate ambiguous target descriptions by looking up the summarization hash map for template matching in constant time. A query may result in one or several valid target descriptions, in which case other miscellaneous information such as camera ID, last seen route, and plate number could be used to narrow the search down to a more specific target. This second-level filtration with plate numbers and other miscellaneous information is not covered within the scope of this paper. In the current system, in cases with multiple valid target descriptions generated by the hashgraph text processing algorithm, the recognition and continuous localization systems will be prompted to track all valid matching targets. The following sample input document will be translated to the equivalent target description using the hashgraph algorithm as follows: "I saw a suspect get into a black car and escaping through the I-93 road. I believe the suspect was driving a Prius, most likely a 2014 model. I saw the car take exit 20. Maybe the color was dark grey I can't be too sure" translates to "Toyota Prius 2014, black or grey", which tells the localization algorithm to search for any Toyota Prius vehicle, 2014 model, with either black or grey color.

In cases of incomplete target descriptions such as "Prius 2012", "Camry 2015", "black BMW", "white Benz", etc., the hashgraph algorithm learns to autocomplete such target descriptions to generate missing label or compute broader scope target description. For example, "Prius 2012" will be computed as "Toyota Prius 2012, any/all color" which means to locate all Toyota Prius 2012 models irrespective of color. Likewise, a target such as "White Benz" will be recomputed as "Mercedes Benz, any/all model, any/all year, white" indicating that the search algorithm should locate all Mercedes Benz, irrespective of model, year, or make but of color white. "Cx7 2009, grey/black" will be computed into "Mazda Cx7 2009, grey or black".

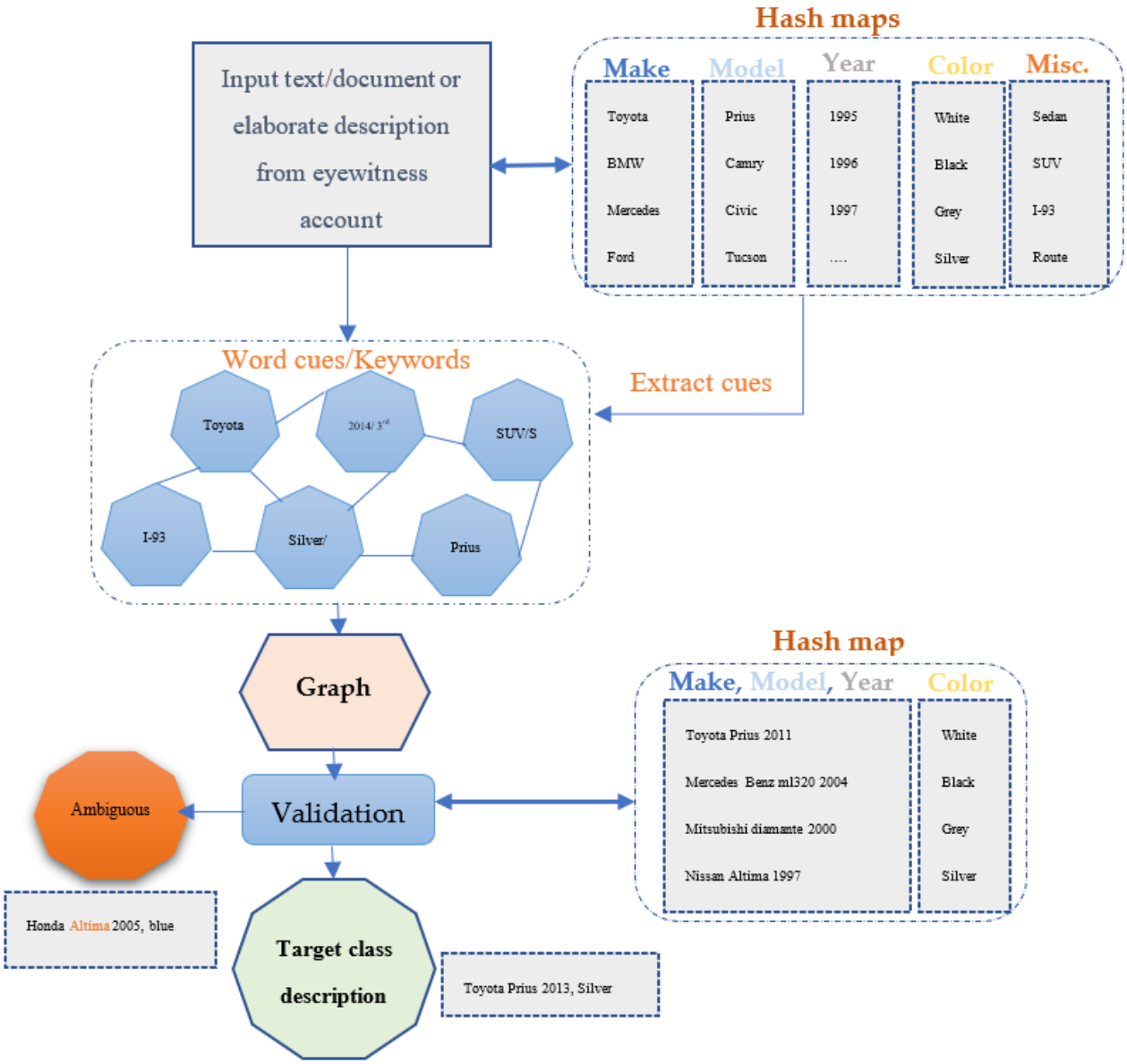

Figure 9. Hashgraph flow diagram for extracting target class description from input text.

4. VRiV Video Testbench Dataset and Proposed Evaluation Metrics

This section of the paper provides details about (a) the proposed VRiV testbench dataset for vehicle recognition in video and (b) the proposed robust evaluation metrics for assessing algorithmic efficiency on recognition in video tasks.

4.1. VRiV (Vehicle Recognition in Video) Dataset

The proposed Vehicle Recognition in Video (VRiV) dataset is the first of its kind and is aimed at developing, improving, and analyzing performance of vehicle search and recognition models on live videos. The lack of such a dataset has limited performance analysis of modern fine-grain vehicle recognition systems to only still image input data, making them less suitable for video applications. The VRiV dataset is introduced to help bridge this gap and foster research in this direction.

The proposed VRiV dataset consists of up to 47 video sequences averaging about 38.5 s per video. The videos are recorded in a traffic setting focusing on vehicles of volunteer candidates whose ground-truth make, model, year, and color information are known. For security reasons and the safety of participants, experiments were conducted on

streets/road with low traffic density. For each video, there is a target vehicle with known ground-truth information, and there are other vehicles either moving in traffic or parked on side streets, to simulate real-world traffic scenario. The goal is for the algorithm to be able to search, recognize, and continuously localize just the specific target vehicle of interest for the corresponding video based on the search query. It is worth noting that the ground-truth information about other vehicles in the videos are not known.

The 47 videos in the testbench dataset are distributed across seven distinct makes and 17 model designs as shown in Figure 10. The videos are also annotated to include groundtruth bounding boxes for the specific target vehicles in the corresponding videos. The dataset includes more than 46 k annotated frames averaging about 920 frames per video. This dataset will be made available on Kaggle, and new videos will be added as they become available.

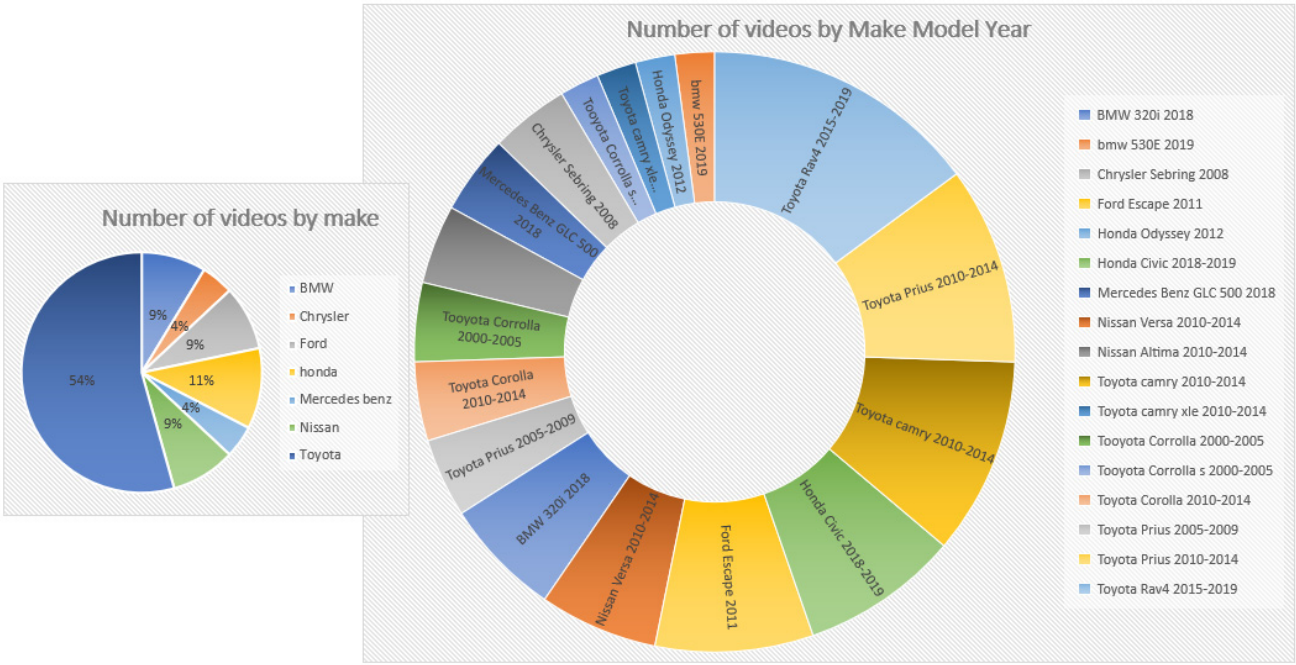

Figure 10. Distribution of the number of video samples in the testbench video dataset for vehicle search localization and tracking. The pie-chart on the left shows the distribution based on the vehicle make, while the larger pie-chart shows the video distribution based on model design and year.

4.2. Evaluation Metrics

No metrics currently exist for quantitatively evaluating the performance of recognition and continuous localization algorithms in video data. The closest metric is the IoU (Intersection of Union), which is used to access the performance of an object tracking algorithm by measuring the overlap ratio between the predicted and groundtruth bounding boxes in each video frame. However, this metric is not well-suited for the task at hand as it does not account for other details such as: how long it takes the model to localize the target once it initially appears in the camera's field of view; how often the model loses track of the target and re-localizes it again; or how close/far away the target has to be for the model to localize it more efficiently; how often the model localizes the wrong target; and how often the model fails to localize a present target.

To measure these attributes, novel metrics are introduced namely: False Localization Rate (*FLR*); Missed Localization Rate (*MLR*); Average Time to First Detection (*ATFD*); Average Continuous Accurate Localization Rate (*ACALR*); and Average Minimum Required Aspect Ratio for accurate detection (*AMRR*).

Additionally, we also consider other metrics such as the overall accuracy of the finer-grain recognition model in a multiclass multi-label setting to comprehensively access the robustness and efficiency of the proposed framework as a whole.

Unlike multiclass single-label classification models, the multiclass multi-label classification models use Exact Matching Ratio to represent the accuracy of the model on still input image data. The Exact Matching Ratio represents the percentage of predictions where all class labels for a given sample perfectly match the corresponding groundtruth labels. This can be computed as:

$$\textit{Exact Matching Ratio, } MR = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{3}$$

where n is the total the number of samples, and I is the perfect matching indicator function, while Y_i and Z_i represent the ground truth and predicted labels, respectively.

The exact matching ratio is considered a harsh metric because it does not consider partially correct predictions. As such, Godbole et al. [48] define the accuracy metric for a multiclass multi-label classification problem as:

$$\textit{Overall Accuracy} = \frac{1}{n}\sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{4}$$

The overall accuracy here represents the average accuracy across all label instances, where accuracy for each instance is the percentage of correctly predicted labels to the total number of labels. This is applicable to still image inputs.

In the proposed V-Localize framework, it is equally important to account for how well the Hashgraph text-based search algorithm is able to compute the appropriate target class from the input label, and how the recognition is able to identify exact matches based on query. The localization efficiency is based on the exact matching ratio as defined in Equation (3). The query processing represents the percentage of time that the algorithm correctly extracts the desired target labels from the input document. This can be defined as:

$$\textit{Exact Matching Ratio with Hashgraph in the loop, } MR_h = Q_p * \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i) \tag{5}$$

where Q_p represents the query processing accuracy, and the rest of the equation is as defined in Equation (3). Assuming a perfect query processing power $Q_p = 1$, Equation (5) reduces to (3).

The following define the novel metrics proposed in this paper, and their mathematical formulae:

(**a**) **False Localization Rate** and (**b**) **Missed Localization Rate**: Given that there exist some inherent inter-make or inter-model ambiguities between vehicles of different brands, or between vehicles of same brands but different models, there is always a possibility for the algorithm to localize the wrong target with high confidence. Similarly, the framework could fail to localize an obvious target due to the viewing angle, or other distortions/occlusions, which are expected in real-world traffic scenarios. As such, it is important to measure how often this type of failure might happen. The False Localization Rate (*FLR*) measures how often the algorithm localizes the wrong target, while the Missed Localization Rate (*MLR*) measures how often the algorithm fails to identify a target present in the video. These are computed respectively as follows:

$$FLR = \sum_{j=1}^{K}\sum_{i=1}^{F_j} \frac{L(y_i \neq z_i)}{F_i} = \frac{\textit{Number of frames where ``wrong target localized''}}{\textit{Total number of frames in testbench dataset}} \tag{6}$$

$$MLR = \sum_{j=1}^{K} \sum_{i=1}^{F_i} \frac{1(P(z_i = 0 \mid y_i = 1))}{\sum_j P(y_i = 1)} = \frac{Number\ of\ frames\ where\ target\ present\ but\ "Not\ localized"}{Total\ number\ of\ frames\ in\ testbench\ dataset\ "where\ target\ present"} \quad (7)$$

where K is the total number of video sequences in the testbench, F_j is the total number of frames in the jth video sequence; $L(y_i \neq z_i)$ is the localization function, which indicates that the target found does not match the known target; $P(z_i = 0 \mid y_i = 1)$ is the probability that a target is not found in the ith frame of the jth video, knowing that the target is present in that frame; and $\sum_{F_j} P(y_i = 1)$ is the total number of frames where the known target is present in a given video.

(**c**) **Average Time to First Detection** (***ATFD***): measures the average time that the algorithm takes to correctly identify a specific target in the video upon a preemptive search request. This helps measure how long, on average, the target continues to go unnoticed/undetected after a search request is entered.

$$ATFD = \frac{1}{K} \sum_{j}^{K} \sum_{i=1}^{F_j} \frac{1(P(z_i = 0 \mid y_i = 1)) * P(L(z_i) = 0)}{\sum_j P(y_i = 1)}, \quad (8)$$

which in simpler notation can be interpreted as:

$$ATFD = \frac{1}{K} \sum_{j}^{K} \left(\frac{Number\ of\ frames\ where\ target\ present\ until\ it\ is\ "localized\ the\ first\ time"}{Total\ number\ of\ frames\ in\ jth\ video\ where\ target\ present} \right) \quad (9)$$

(**d**) **Average Continuous Accurate Localization Rate** (***ACALR***): measures the average amount of time that a specific target is successfully localized, consistently, across frames of a video sequence. Considering a random video V_j consisting of 500 frames, in which a target T_i is apparent for 100 frames total (for example, from frame 250 to 350), the continuous accurate localization for that video will be defined as the total number of frames (between frame 250 and 350 where the target is present) where the algorithm successfully localizes the target.

$$ACALR = \frac{1}{K} \sum_{j}^{K} \sum_{i=1}^{F_j} \frac{1\ (L(z_i = y_i) * P(y_i = 1)\ )}{\sum_{F_j} P(y_i = 1)} \quad (10)$$

$$ACALR = \frac{1}{K} \sum_{j}^{K} CALR \quad (11)$$

$$CALR = \frac{Number\ of\ frames\ where\ target\ present\ and\ successfully\ localized}{Total\ number\ of\ frames\ where\ target\ present\ in\ the\ video} \quad (12)$$

where $CALR$ represents the Continuous Accurate Localization Rate for a single video where the target is present.

(**e**) **Average Minimum Required Aspect Ratio** (***AMRR***) **for accurate detection** measures how apparent the target must be in the video before the algorithm can detect it. It is defined as the ratio between the surface area of the smallest successfully predicted bounding box for a specific target, and the surface area of the entire frame. In other words, it is a measure of how well the algorithm can see clearly and identify targets accurately across different scales.

$$AMRR = \frac{1}{K} \sum_{j}^{K} \frac{Min\Big(A_{xywh}(L(z_i = y_i) \Big)}{A_{XYWH}} \quad (13)$$

where K is the total number of videos in the dataset with valid target present, $Min\Big(A_{xywh}(L(z_i = y_i) \Big)$ is the minimum bounding box area of a correctly localized target

in a given video, and A_{XYWH} is the total area of a single frame from a given video, assuming consistent frame dimensions across frame sequences in a given video.

We argue that the proposed evaluation metrics are more adequate metrics for reporting the effectiveness and robustness of vehicle recognition algorithms in videos more efficiently. Evaluation results from computer simulations are detailed in Section 5.

5. Experimental Results and Discussions

The performance of the VMMYC vehicle recognition model was compared with different backbone architectures, particularly, ResNet, RepVGG, MobileNetv3, and VGG16. Table 1 reports the top one and top three accuracy on the MMY (Make Model Year) and color predictions on still images from a validation subset of the curated VMMR dataset. Accuracies reported in Table 1 are for individual labels.

Table 1. Comparative performance of various models on the vehicle recognition task using still images from the refined VMMR dataset.

	Make Model Year		**Color**	
	Top 1 Accuracy (%)	**Top 3 Accuracy (%)**	**Top 1 Accuracy (%)**	**Top 3 Accuracy (%)**
ResNet-CBAM	83.28	97.42	90.68	96.95
ResNet50	82.40	95.23	89.22	95.12
MobileNetV3	76.72	90.72	87.65	93.3
RepVGG	85.59	98.10	91.60	97.54
VGG16	75.27	90.15	85.29	93.11

The overall multiclass multi-label recognition accuracy for various architectures are presented in Table 2 and were computed using Equation (2).

Table 2. Overall recognition accuracy for the Vehicle Make Model Color.

	Make Model Year Color	
	Top 1 Accuracy (%)	**Top 3 Accuracy (%)**
ResNet-CBAM	86.823	97.184
ResNet50	85.674	95.175
MobileNetV3	81.822	91.992
RepVGG	88.493	97.819
VGG16	79.967	91.606

The results presented in Tables 1 and 2 suggest that RepVGG and ResNet-CBAM yielded the highest recognition accuracy. However, RepVGG proves more computationally efficient in terms of the required recognition speed necessary for the instant search and localization task. RepVGG achieved comparable and even better results than ResNet and ResNet-CBAM with half as many trainable parameters, which makes the RepVGG backbone very lightweight and much faster, and therefore most suitable for the application at hand.

The robustness and effectiveness of the proposed vehicle search, recognition, and continuous localization framework was further evaluated on the VRiV dataset, which contained vehicles with known ground-truth about their make, design, year, and color information as well as their positional coordinates (bounding boxes) in the video sequence.

First, the False Localization Rate (*FLR*), which measures how often the algorithm localizes a wrong target; and the Missed Localization Rate (*MLR*), which measures how often the algorithm fails to localize a target that is apparent in the video sequence were

computed. The results shown in Table 3 indicate that the algorithm localized the wrong target only **0.03%** of the time (*FLR*). This result was based on the average across all 47 videos in the VRiV testbench. The *MLR* on the other hand was at about **26.22%,** indicating that the model failed to localize a present target about **26%** of the time. This could be due to several factors including but not limited to distance of the target from the camera, partial occlusions, quality of the video feed, environmental conditions, and color mismatch.

Table 3. Overall performance evaluation of the proposed novel Vehicle Search, Recognition, and Continuous Localization framework on the proposed VRiV testbench video dataset using the proposed novel video recognition evaluation metrics.

Performance Metrics for V-Localize (Search, Recognition and Continuous Localization Framework)				
***AMMR* (%) ↓**	***FLR* (%) ↓**	***MLR* (%) ↓**	***ATFD* ↓**	***CALR* ↑**
3.23	0.03	26.22	0.268	0.262

Additionally, Table 3 reports the results of the corresponding *AMMR*, *ACALR*, and *ATFD* on the VRiV dataset. The *AMMR*, which measures the average aspect ratio between the localization bounding box and the frame size indicates how small or large the target of interest must be, with respect to the video frame size, for the algorithm to successfully localize it.

The average *AMMR* for all 47 videos was evaluated at **3.23%** indicating that the algorithm could recognize fairly small targets, which were quite far away from the camera focal point. Similarly, the *ACALR* or simply *CALR* measures how consistently the algorithm can accurately and continuously localize and track a moving target in a video sequence. This metric also indicates the average percentage of the time that a moving target will be successfully localized and recognized in a given video sequence. A *CALR* of **0** indicates that the target completely went unnoticed, and a *CALR* of **1** indicates that the target was continuously localized **100%** of the time as it moves across the frames. The Average *CALR* for the proposed testbench was computed as **0.262**.

Sample results for the vehicle search and localization are shown in Figure 11. The tags below each image represent simplified or preprocessed text-input queries. The red circles are the corresponding localizations based on the input queries. These are based on extracts from the VRiV testbench. Localized targets match the corresponding ground-truth based on input text query.

Further experimental results presented in Table 4 show the comparative performance of the several backbone architectures used in the proposed V-Localize framework. These results suggest that the RepVGG backbone performed better than all other recognition backbones on all metrics. The ResNet-CBAM model was shown to yield a slightly better *AMMR* than RepVGG but within the margin of error. Arrows are used in Tables 3 and 4 to facilitate score interpretability for the corresponding metrics. A downward arrow suggests that the smaller the score, the better the performance of the model, while an upward arrow suggests that a higher score indicates better performance.

Figure 11. Sample results for vehicle search and localization. The tags below each image represent simplified or preprocessed text-input queries. The red circles are the corresponding localizations based on the input queries. Sample results shown here are based on extracts from the video VRiV testbench dataset, and the localizations match the corresponding ground truth based on query.

Table 4. Comparative performance analysis of various backbone architecture in the V-Localize framework using the proposed evaluation metrics. The arrows next to each metric suggest if a higher or lower score indicates better performance index. Downward arrow means the lower the score, the better the model, while an upward arrow indicates that the higher the score, the better the model.

	Performance Metrics for V-Localize (Search, Recognition and Continuous Localization Framework)				
BACKBONE	***AMMR* (%) ↓**	***FLR* (%) ↓**	***MLR* (%) ↓**	***ATFD* ↓**	***CALR* ↑**
RepVGG	3.23	**0.03**	**26.22**	**0.268**	**0.262**
ResNet-CBAM	**3.21**	0.08	29.55	0.295	0.253
ResNet50	3.28	0.07	30.69	0.294	0.249
MobileNetV3	3.47	0.11	35.22	0.314	0.223
VGG16	4.81	0.17	36.71	0.310	0.206

The bold indicates the highest performing for corresponding metric.

6. Conclusions and Future Work

In this paper, we addressed the problem of vehicle search and continuous localization in traffic videos which has yet to be explored extensively in the scholarly literature. A sophisticated artificial intelligence framework, **V-Localize**, is proposed to solve this by achieving finer-grain recognition in live traffic video. The concept of text-based search is explored by introducing a novel ***Hashgraph*** algorithm to efficiently process input documents and compute valid target descriptions, which are used to trigger a query of the continuous localization model.

An important differentiating factor of the finer-grain vehicle recognition (*VMMYC*) component of the V-Localize framework is that beyond predicting the make, model, and year of a given vehicle, it also predicts the color information. To achieve this, the most diverse and large-scale vehicle color dataset, **VCoR,** was created, which includes **15** of the most popular vehicle color classes, representing twice as much color diversity as there is in the largest existing such dataset, making it more suitable for finer-grain recognition. The **VCoR** dataset also contains over **10k** image samples. Several architectures were explored for the VMMYC module, and **RepVGG** was experimentally adopted as the backbone for the proposed multiclass multiple-label classification model to recognize vehicles by their make, model, year, and color information respectively. The RepVGG model trained on the curated VMMR dataset as detailed in the paper, achieved a top one classification accuracy of **88.49%** and top three accuracy of **97.83%** on the still images data.

Additionally, a first of its kind testbench video dataset, the **VRiV** (Vehicle Recognition in Video) was created to facilitate proper performance evaluation of vehicle recognition models on live-video feed. The reason modern vehicle recognition models only report accuracy on still imagery could be attributed to the lack of such a dataset. The proposed **VRiV** dataset consists of **47 video sequences** in live traffic, each containing visual data of a vehicle with known ground-truth data about its model, make, year, and color. The testbench contains over **46k** annotated frames averaging about **38.5** s and **920 frames** per video sequence. The **VRiV** dataset is meant to steer development of more advancement recognition models well-suited for video applications.

Lastly, we proposed novel metrics including ***FLR***, ***MLR***, ***AMMR***, ***ACALR***, and ***ATFD*** to address the lack of such metrics in the existing literature, and to foster proper development and quantitative assessment of robustness and performance efficiency of recognition models in video data. Quantitative performance evaluation of the proposed framework was performed on the **VRiV** testbench, as reported and analyzed in result section. The proposed metrics could also prove very useful in other recognition applications.

One major advantage of the proposed system is that it can be integrated into intelligent transportation system software to aid law enforcement in fighting crimes on highway roads.

Future work will be directed at expanding the **VRiV** dataset to include more challenging videos and increasing the diversity of target vehicle in terms of models, make, year, and color. Additionally, more work needs to be completed to improve the recognition accuracy as well as the AMMR and average CALR of the localization and recognition framework. One way to achieve this will be to develop a novel backbone neural net architecture that is better-suited for finer-grain recognition in live videos.

Author Contributions: Conceptualization, K.P., L.K., and V.O.; methodology, L.K., V.O. and K.P.; software, K.P., L.K., V.O.; validation, K.P., L.K., V.O., J.I., S.A.; formal analysis, K.P., L.K., S.A., J.I.; investigation, L.K., V.O., Karen Panetta; resources, K.P.; data curation, L.K.; writing—original draft preparation, L.K., V.O., K.P.; writing—review and editing, K.P., L.K., S.A., V.O., J.I.; visualization, L.K.; supervision, K.P., J.I., S.A.; project administration, K.P., J.I., S.A.; funding acquisition, K.P., L.K., V.O., J.I., and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the US Department of Transportation, Federal Highway Administration (FHWA), grant contract: 693JJ320C000023.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of Tufts University (protocol title: Tufts Driving Dataset; protocol code: STUDY00001272; date of approval: 3 June 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study (VRiV dataset collection).

Data Availability Statement: Datasets proposed in this paper are made publicly available on Kaggle. VCoR: https://www.kaggle.com/landrykezebou/vcor-vehicle-color-recognition-dataset. VRiV: https://www.kaggle.com/landrykezebou/vriv-vehicle-recognition-in-videos-dataset (accessed on 2 December 2021).

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

References

1. Manzoor, M.A.; Morgan, Y.; Bais, A. Real-Time Vehicle Make and Model Recognition System. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 611–629. [CrossRef]
2. Vehicle Make Model Recognition with Color | Plate Recognizer ALPR. Available online: https://platerecognizer.com/vehicle-make-model-recognition-with-color/ (accessed on 5 April 2021).
3. Chan, F.-H.; Chen, Y.-T.; Xiang, Y.; Sun, M. Anticipating Accidents in Dashcam Videos. In Proceedings of the 13th Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 136–153. [CrossRef]
4. Shah, A.P.; Lamare, J.-B.; Nguyen-Anh, T.; Hauptmann, A. CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis. In Proceedings of the 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–9. [CrossRef]
5. Sultani, W.; Chen, C.; Shah, M. Real-World Anomaly Detection in Surveillance Videos. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6479–6488. Available online: http://crcv.ucf.edu/projects/real-world/ (accessed on 5 April 2021).
6. Siddiqui, A.J.; Mammeri, A.; Boukerche, A. Real-Time Vehicle Make and Model Recognition Based on a Bag of SURF Features. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3205–3219. [CrossRef]
7. Kezebou, L.; Oludare, V.; Panetta, K.; Agaian, S.S. Joint image enhancement and localization framework for vehicle model recognition in the presence of non-uniform lighting conditions. In *Multimodal Image Exploitation and Learning 2021*; International Society for Optics and Photonics: Bellingham, DC, USA, 2021. [CrossRef]
8. Kezebou, L.; Oludare, V.; Panetta, K.; Agaian, S.S. A deep neural network approach for detecting wrong-way driving incidents on highway roads. In *Multimodal Image Exploitation and Learning 2021*; International Society for Optics and Photonics: Bellingham, DC, USA, 2021. [CrossRef]
9. Tafazzoli, F.; Frigui, H.; Nishiyama, K. A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 874–881. [CrossRef]
10. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for Fine-Grained Categorization. In Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 554–561.
11. Yang, L.; Luo, P.; Loy, C.C.; Tang, X. A large-scale car dataset for fine-grained categorization and verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3973–3981. [CrossRef]
12. Boonsim, N.; Prakoonwit, S. Car make and model recognition under limited lighting conditions at night. *Pattern Anal. Appl.* **2017**, *20*, 1195–1207. [CrossRef]
13. Pearce, G.; Pears, N. Automatic make and model recognition from frontal images of cars. In Proceedings of the 2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, Klagenfurt, Austria, 30 August–2 September 2011; pp. 373–378. [CrossRef]
14. Lee, H.J.; Ullah, I.; Wan, W.; Gao, Y.; Fang, Z. Real-Time Vehicle Make and Model Recognition with the Residual SqueezeNet Architecture. *Sensors* **2019**, *19*, 982. [CrossRef] [PubMed]
15. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
16. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Computer Vision—ECCV 2006, Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417. [CrossRef]
17. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
18. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005. [CrossRef]
19. Suykens, J.A.K.; Vandewalle, J. Least Squares Support Vector Machine Classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
20. Support Vector Machines—Scikit-Learn 0.24.2 Documentation. Available online: https://scikit-learn.org/stable/modules/svm.html (accessed on 15 May 2021).
21. Sklearn.cluster.k_means—Scikit-Learn 0.24.2 Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.k_means.html?highlight=kmeans#sklearn.cluster.k_means (accessed on 15 May 2021).
22. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [CrossRef]
23. Sklearn.Ensemble.RandomForestClassifier—Scikit-Learn 0.24.2 Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed on 15 May 2021).
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

26. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5 MB Model Size, February 2016. Available online: http://arxiv.org/abs/1602.07360 (accessed on 6 April 2021).
27. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. PMLR. 2019. Available online: http://proceedings.mlr.press/v97/tan19a.html (accessed on 13 May 2021).
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Available online: http://www.robots.ox.ac.uk/ (accessed on 6 April 2021).
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 779–788. Available online: http://arxiv.org/abs/1506.02640 (accessed on 6 April 2021).
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37. [CrossRef]
31. Kazemi, F.M.; Samadi, S.; Poorreza, H.R.; Akbarzadeh, T.M.-R. Vehicle Recognition Using Curvelet Transform and SVM. In Proceedings of the Fourth International Conference on Information Technology (ITNG'07), Las Vegas, NV, USA, 2–4 April 2007; pp. 516–521. [CrossRef]
32. Manzoor, M.A.; Morgan, Y. Vehicle Make and Model classification system using bag of SIFT features. In Proceedings of the th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 9–11 January 2017; pp. 1–5. [CrossRef]
33. Wang, H.; Yu, Y.; Cai, Y.; Chen, L.; Chen, X. A Vehicle Recognition Algorithm Based on Deep Transfer Learning with a Multiple Feature Subspace Distribution. *Sensors* **2018**, *18*, 4109. [CrossRef] [PubMed]
34. Fang, J.; Zhou, Y.; Yu, Y.; Du, S. Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1782–1792. [CrossRef]
35. Ma, X.; Boukerche, A. An AI-based Visual Attention Model for Vehicle Make and Model Recognition. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6. [CrossRef]
36. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
38. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. *CBAM: Convolutional Block Attention Module*; Springer: Singapore, 2018; Volume 11211, pp. 3–19. Available online: http://arxiv.org/abs/1807.06521 (accessed on 15 May 2021).
39. Understanding Random Forest. How the Algorithm Works and Why it Is . . . | by Tony Yiu | Towards Data Science. Available online: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed on 15 May 2021).
40. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Hartwig, A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. Available online: http://arxiv.org/abs/1704.04861 (accessed on 6 April 2021).
41. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2021; pp. 13728–13737. [CrossRef]
42. Chen, P.; Bai, X.; Liu, W. Vehicle Color Recognition on Urban Road by Feature Context. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2340–2346. [CrossRef]
43. Rachmadi, R.F.; Purnama, I.K.E. Vehicle Color Recognition Using Convolutional Neural Network. *arXiv* **2015**, arXiv:1510.07391. Available online: http://arxiv.org/abs/1510.07391 (accessed on 19 April 2021).
44. Li, X.; Zhang, G.; Fang, J.; Wu, J.; Cui, Z. Vehicle Color Recognition Using Vector Matching of Template. In Proceedings of the Third International Symposium on Electronic Commerce and Security, Nanchang, China, 29–31 July 2010; pp. 189–193. [CrossRef]
45. Hu, C.; Bai, X.; Qi, L.; Chen, P.; Xue, G.; Mei, L. Vehicle Color Recognition With Spatial Pyramid Deep Learning. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2925–2934. [CrossRef]
46. Which Car Color Is Most Popular in Your State? | CarMax. Available online: https://www.carmax.com/articles/car-color-popularity (accessed on 16 May 2021).
47. CarMax—Browse Used Cars and New Cars Online. Available online: https://www.carmax.com/ (accessed on 16 May 2021).
48. Godbole, S.; Sarawagi, S. Discriminative Methods for Multi-labeled Classification. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 26–28 May 2004; pp. 22–30. [CrossRef]