



Research article

Automated vehicle damage classification using the three-quarter view car damage dataset and deep learning approaches

Donggeun Lee ^a, Juyeob Lee ^b, Eunil Park ^{b,c,*}

^a Department of Artificial Intelligence, Sungkyunkwan University, Seoul 03063, Korea

^b Department of Applied Artificial Intelligence, Sungkyunkwan University, Seoul 03063, Korea

^c Teach Company, Seoul 03063, Korea

ARTICLE INFO

Dataset link: <https://github.com/dxlabskku/TQVCD.git>

Keywords:

Vehicle damage
Damage classification
Neural network
Deep learning
Model ensemble
Transfer learning

ABSTRACT

Automated procedures for classifying vehicle damage are critical in industries requiring extensive vehicle management. Despite substantial research demands, challenges in the field of vehicle damage classification persist due to the scarcity of public datasets and the complexity of constructing datasets. In response to these challenges, we introduce a Three-Quarter View Car Damage Dataset (TQVCD dataset), emphasizing simplicity in labeling, data accessibility, and rich information inherent in three-quarter views. The TQVCD dataset distinguishes class by vehicle orientation (front or rear) and type of damage while maintaining a three-quarter view. We evaluate performance using five prevalent pre-trained deep learning architectures—ResNet-50, DenseNet-160, EfficientNet-B0, MobileNet-V2, and ViT—employing a suite of binary classification models. To enhance classification robustness, we implement a model ensemble method to effectively mitigate individual model dependencies' deviations. Additionally, we interview three experts from the used-car platform to validate the necessity of a vehicle damage classification model using the corresponding dataset from an industrial perspective. Empirical findings underscore the dataset's comprehensive coverage of vehicle perspectives, facilitating efficient data collection and damage classification while minimizing labor-intensive labeling efforts.

1. Introduction

Used-car platforms, car-sharing platforms, and car insurance companies require swift and efficient procedures to handle large-scale vehicle fleets. In particular, an automated process for classifying vehicle damage is indispensable for the practical management of many accident-involved vehicles. This automated system ensures precise vehicle damage classification and management, facilitating more efficient claims processing and insurance procedures [33].

Moreover, the automated classification process minimizes the potential for errors and promotes fairness by delivering accurate results, thereby minimizing reliance on subjective human judgment. In addition, it contributes to both time and cost savings, reducing processing time and improving the overall efficiency of the claims procedures. Reducing necessary downtime for vehicle repair and maintenance leads to not only time-saving and improvements in vehicle availability but can also enhance customer satisfaction for vehicle insurance companies and elevates vehicle maintenance and service quality for used-car platforms and car-sharing platforms [24].

* Corresponding author at: 310 International Hall, Sungkyunkwan University, 25-2 Sungkyunkwan-ro, Jongno-gu, Seoul 03063, Korea.
E-mail address: eunilpark@skku.edu (E. Park).

<https://doi.org/10.1016/j.heliyon.2024.e34016>

Received 25 February 2024; Received in revised form 15 June 2024; Accepted 2 July 2024

Available online 8 July 2024

2405-8440/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

However, addressing a vehicle damage classification task with real-world data poses some notable challenges that demand attention. First, car-related damage datasets are rarely public. Second, there is no dataset containing various environmental conditions and lighting scenarios. Third, collecting and labeling fine-grained data one by one is complicated and inefficient. These issues not only delay the progress of AI in automotive applications but also limit the potential for widespread adoption and optimization of such technologies in the sector.

Singh et al. [33] classified the degree of vehicle damage into five levels and utilized Mask R-CNN for damage detection and segmentation across 32 vehicle parts. Balci et al. [4] proposed a dataset, which was curated specifically for the frontal vehicle viewpoints, encompassing instances of both damaged and undamaged conditions. Using convolutional network networks (CNNs), feature extraction was conducted on the dataset. Subsequently, support vector machines (SVMs) were utilized for binary classification with extracted features.

Waqas et al. [39] investigated that vehicle damage severity was classified into three levels: none, medium, and large. MobileNet was employed for the classification task to evaluate and categorize the severity of vehicle damage across distinct levels. Kyu and Woraratpanya [22], developed a comprehensive pipeline for the classification of damage detection. However, performance was comparatively lower for damage location detection, achieving an accuracy of 70%. Similarly, the classification accuracy for damage severity was around 57.8%, indicating a lower efficacy in this specific aspect of the pipeline.

Artan and Kaya [3] developed classifiers using Inception V3 and MobileNet to categorize damage severity for bumper, hood, and fender parts of cars into three distinct levels. Precision values for damage classification exhibited a range of performance, with a maximum precision of 90% and a minimum precision of 47%. These results highlight the variability in the efficacy of the classifiers for distinguishing different levels of damage across specified vehicle parts.

Chua et al. [7] conducted a damage classification task using Convolutional Neural Networks (CNNs) specifically for three car parts: front bumper, rear bumper, and car wheel. However, the validation accuracy for this classification task was reported at 49.28%. Notably, the test dataset employed for evaluation consisted of only five images per class, indicating a limited sample size that may not robustly demonstrate the model's performance on unseen data. Moreover, Dwivedi et al. [11] employed YOLOv3 for examining damage detection tasks. Subsequently, a CNN-based classification model, specifically ResNet50, achieved an impressive accuracy of 96.39% for classifying eight distinct classes. However, it's important to note that the reported test results in this study were based on augmented data, potentially limiting their representation of real-world scenarios due to the augmentation techniques applied.

Thus, we introduce novel approaches that aim to address current challenges in the vehicle damage classification task. The main contributions are presented as follows:

- We introduce a novel vehicle damage dataset, the Three-Quarter View Car Damage dataset (TQVCD dataset). The main objective of the dataset is to efficiently acquire comprehensive information on damage without explicitly labeling every damaged part.
- The TQVCD dataset presents class criteria considering vehicle orientation and type of damage, unlike other datasets. In addition, images were collected for four vehicle sizes to create a vehicle-size robust dataset.
- Our research is expected to be the basis for future research on the automobile industry with public data. Most studies have different criteria for each study using their own datasets, and there are few public car damage datasets.
- To validate our dataset, we leverage transfer learning with diverse pre-trained deep learning-based models to build a binary classifier for each type of damage. To improve the robustness of classification results, we implemented a model ensemble method to reduce the deviation of individual model dependencies effectively. Our experiments included various weight values for every single model to optimize classifier performance.
- In recognition of the reduced sensitivity to lighting and background noise, we incorporated grayscale datasets into our analysis. The final results were derived by combining prediction values obtained from RGB datasets and computing the elementwise average.
- Since it is an industrial issue, the effectiveness of this study was verified through interviews with industry experts, and the effectiveness of the paper was reviewed.

The remainder of the paper is structured as follows: Section 2 delves into an exploration of prior research on vehicle-related datasets with several classification tasks. Section 3 introduces the TQVCD dataset. Then, the proposed framework for classifying vehicle damage types, including various backbone networks and evaluation metrics, is presented in Section 4. Both Sections 5 and 6 present our experimental setup and results, respectively. Section 6 includes the results of reviewing the effectiveness of the study to industry experts in the field who manage large vehicles on used car platforms. Concluding our research, Section 7 provides a summary of our findings and suggests future work.

2. Related work

2.1. Vehicle-related datasets

Krause et al. [21] introduced the Stanford-Cars dataset, a large-scale dataset of car models that contains 16,185 images of 197 classes. Each image was sourced from websites focusing on specific vehicle manufacturing brands. Similarly, Yang et al. [40] presented the CompCars dataset, another large-scale dataset featuring diverse car views, comprehensive coverage of internal and external parts, and rich attributes. The dataset encompasses a total of 136,722 images capturing entire cars and 27,618 images specifically focusing

Table 1

Comparison of vehicle damage datasets. In the “Task” column, ‘C’ stands for Classification, ‘D’ for Detection, and ‘S’ for Segmentation. The “# Data” column represents the number of vehicle images available in each dataset. If the quantity is not specified in the paper, “N/A” is indicated. The “# Cat” column denotes the number of categories present in the dataset. The “Category Name” column describes the specific categories utilized by each dataset. The “Dmg” column indicates whether the dataset encompasses vehicle damage. Lastly, the “Public” column signifies whether the data is publicly accessible. It’s noteworthy that there are only a few publicly available datasets for vehicle damage analysis.

Dataset	Task	# Data	# Cat	Category Name	Dmg	Public
Patil et al. [28]	C	1,503	7	bumper dent, door dent, glass shatter, headlamp broken, tail lamp broken, scratch, smash	✓	×
De Deijn [8]	C	1,007	4	dent, glass, hail, scratch	✓	×
Li et al. [24]	D	1,790	3	scratch, dent, crack	✓	×
Balci et al. [4]	C	533	2	damaged, non-damaged	✓	×
Dhieb et al. [9]	D, S	N/A	3	minor, moderate, major	✓	×
Singh et al. [33]	S	2,822	5	scratch, major dent, minor dent, cracked, missing	✓	×
Waqas et al. [39]	C	600	3	medium damage, huge damage, no damage	✓	×
Patel et al. [27]	D	326	3	bump, dent, scratch	✓	×
Dwivedi et al. [11]	C, D	1,077	7	bumper dent, door dent, glass shatter, headlamp broken, tail lamp broken, scratch, smash	✓	×
Chua et al. [7]	C	500	3	front bumper, rear bumper, car wheel	✓	×
Qaddour and Siddiqua [29]	C	N/A	3	minor damage, moderate damage, severe damage	✓	×
Seo et al. [31]-Defect	C	2,000	4	scratch, dents, spacing, breakage	✓	×
Wang et al. [38]	C, D, S	4,000	6	dent, scratch, crack, glass shatter, lamp broken, tire flat	✓	✓
TQVCD	C	2,300	6	front breakage, front crushed, rear breakage, rear crushed, front normal, rear normal	✓	✓

on car parts. These images were sourced from websites and surveillance cameras. While these datasets excel in addressing fine-grained tasks related to categorizing cars and their components, it’s crucial to note that they are not designed for car damage classification.

Most studies on compromised vehicles tend to generate and analyze their datasets, with very little public data. Balci et al. [4] contributed a dataset focused on front-view vehicles, specifically aimed at determining damage. The dataset contained 533 instances of damaged images, all about the vehicle’s front view. However, it is important to note that this dataset is limited to front-view scenarios, and scalability in other directions is a primary concern.

Patil et al. [28] introduced a distinctive damaged vehicle dataset comprising seven classes. They utilized websites to facilitate the multi-classification of damage. The dataset contains a total of approximately 1,503 damaged image instances, but there is a significant imbalance across classes ranging from 100 to 269. De Deijn [8] constructed dataset by considering various types, locations, and sizes of vehicle damage. However, the dataset was built without considering different models, considering only two vehicles popular in the country.

Chua et al. [7] developed a dataset with three classes: front bumper, rear bumper, and wheels. A limitation of this dataset is that it only allows observation of the bumper or wheel of the vehicle. These datasets are not sufficient to validate with deep learning. Singh et al. [33] presented a car damage dataset that includes 2,822 images sourced from a database of insurance claims. The dataset is comprehensively annotated with a detailed description of the damaged area, but it does not describe the direction of each damage category and lacks information about the diversity of vehicle size.

Waqas et al. [39] collected a total of 600 images consisting of three classes without moderate damage, major damage, and damage. It was collected from various angles based on the front of the vehicle, but there is no image data about the rear or information about the car model diversity. Patel et al. [27] consisted of 326 images of both damaged and undamaged vehicles, captured using smartphone cameras or sourced from the internet. Although various shapes and intensities of damage were collected, the dataset is small, limited to a specific area, and primarily composed of close-up images.

Dwivedi et al. [11] developed a dataset through web crawling that includes images classified into an undamaged class and seven commonly observed types of damage. The damage categories are bumper dents, scratches, door dents, shattered glass, headlamp damage, taillamp damage, and destruction. Qaddour and Siddiqua [29] constructed two datasets: one consisting of images of cars and various objects and the other containing images of both damaged and undamaged vehicles. The damage in the second dataset was categorized into significant, moderate, and minor. They employed data augmentation techniques such as flipping, zooming, shifting, and random rotation to combat overfitting during training.

Seo et al. [31] establish the SOCAR datasets, which offer real-world car image datasets with more diverse attributes. The SOCAR datasets consist of 10,000 images of 14 classes and also consist of the following three subsets: Main, Defect, and Dirt. Within the SOCAR-Defect subset are 2,000 samples of damage data related to the vehicle’s external surface, encompassing scratches, dents, and breakage.

Wang et al. [38] introduced CarDD datasets, comprising 4,000 high-resolution images depicting car damage, accompanied by annotations for over 9,000 instances across six damage categories. These datasets aim to establish fine-grained datasets, considering various real-world environments. Notably, the complexity and labeling challenges arise due to the presence of numerous intra-classes within the datasets. Table 1 summarizes the existing vehicle-related damage datasets.

2.2. Vehicle-related classification

The utilization of pre-trained models for feature extraction and classification has been a predominant approach in various studies concentrating on car damage detection. Patil et al. [28] explored and implemented transfer learning for CNN-based models and ensemble learning. Their approach yielded an accuracy of 88.24% and an F1-score of 78.41% without employing data augmentation.

Sruthy et al. [34] implemented vehicle damage classification tasks through diverse CNN methods, including Inception-V3, Xception, VGG16, VGG19, ResNet50, and MobileNet. They reported that only MobileNet achieved a noteworthy 92.7% validation accuracy. Similarly, Dwivedi et al. [11] conducted a multi-class task for vehicle damage classification, employing pre-trained CNN models such as Alexnet, VGG-19, Inception V3, MobileNets, and ResNet50. They categorized models into three blocks and applied a differential learning rate for each block. This technique led to a test accuracy of 96.39% for ResNet50. However, it should be noted that this method entails intricate model manipulation and extensive training time, as different learning rates must be applied for each model block.

Artan and Kaya [3] proposed a model for classifying damage based on damage severity across different classes, specifically bumper, hood, and fender. They categorized damage severity into no damage, minor, and severe. Impressively, over 90% precision was achieved for both no damage and bumper classes, while 63.3% precision was presented for the hood class with minor damage.

In a related context, Kyu and Woraratpanya [22] proposed vehicle damage assessment pipelines utilizing the VGG family. The pipeline included three evaluation components: damage detection, location detection, and severity detection. Notably, the most robust performance in damage detection was observed with a VGG19 model, achieving an accuracy of 95.22%. However, in damage location and severity detection, the accuracy figures dropped to 76.48% and 57.89%, respectively, suggesting that accurately classifying minor damage to vehicles remains a challenging task. To address the limitations identified in previous research, our study introduces innovative datasets and simplifies data management and labeling complexities. Subsequently, after the learning phase on our proprietary dataset using transfer learning, our research concentrates on improving the overall model performance on test data through the application of a weighted ensemble technique. Notably, in contrast to earlier studies, we replicate the same process on grayscale datasets, thereby augmenting and complementing the results obtained from RGB datasets.

3. Dataset

Considering the scarcity of publicly available datasets associated with car damages [31,38], we introduce our dataset, providing a unique perspective distinct from existing datasets. The following sections provide comprehensive details on how we define and construct our dataset, showing a novel approach to addressing the limitations observed in other datasets.

3.1. Data configuration and collection

First, we define a collect images of vehicles from a three-quarter view perspective, both for the front and rear of the vehicle. The three-quarter view, a widely utilized perspective in art, photography, and design, involves depicting an object from an angle approximately 45 degrees off-center. This perspective effectively captures three sides or dimensions of the object, providing depth and dimension, and revealing details from an angled viewpoint.

In this specific context, adopting the three-quarter view allows for a comprehensive assessment of accident vehicles within a single image. It goes beyond merely presenting the front, rear, and sides of the car, providing insights into damaged components such as the grille, headlights, and other car parts. When capturing the three-quarter view of the vehicle front, the images capture the front of the car from an angle of about 45 degrees. This angle ensures visibility of the front end, encompassing components like the grille, front fender panel, front bumper, headlights, and other relevant parts. Similarly, for the three-quarter view of the vehicle's rear, images are captured from an angle of about 45 degrees. This angle allows for a view of the rear end, including components such as the trunk, rear fender panel, rear bumper, taillights, and other relevant parts.

The dataset consists of un-damaged and damaged, and the un-damaged data serves as ground truth. Each configuration is collected for the front and rear. The damaged datasets are again divided into breakage and crushed and are collected for front and rear, respectively. In the context of car accidents, 'Breakage' and 'Crushed' are terms used to describe different types of damage to vehicles. "Breakage" typically involves the shattering or breaking of specific components such as windows, bumper, headlights, or other parts of the vehicle. "Crushed" refers to the deformation or compression of the vehicle's structure, often caused by the force of a collision. For each of the damaged images, we consist of both mild and severe damage for the robustness of the dataset. The details of our dataset are shown in Table 2 and Fig. 1.

Fig. 2 shows in pie charts whether vehicle diversity is well reflected when the car model is divided into four categories according to vehicle size within each label. Categories according to vehicle size are classified into sub-compact cars, compact cars, mid-size cars, and full-size cars. The pickup truck and van belong to full-size cars. TQVCD datasets generally show similar distributions for each label, but for sub-compact and compact cars, the number and distribution of data are relatively less in the rear than in the front.

We obtained images of un-damaged vehicles by taking them directly or from sites such as used car platforms. We collected images of damaged vehicles from a vehicle damage dataset in the AI-HUB (AI Hub). This is a dataset for 632,694 accident vehicle images and we selected only the available data from the perspective we define.

Undamaged vehicle images were acquired via direct capture or extraction from websites, including used car platforms. For damaged vehicle images, we utilized a dataset from AI-hub [1], specifically focused on vehicle damage. The AI-hub dataset comprises a substantial collection of 632,694 accident vehicle images. We select the images, which align with the specific perspectives and criteria defined in our research.



Fig. 1. Samples for each label of the TQVCD dataset. Figure presents samples for each type of damage. In the example for each type of damage, the upper two lines of the picture are samples for the front of the vehicle, and the lower two lines are for the rear. The TQVCD dataset is labeled based on the point of view, the orientation of the vehicle, and the type of damage.

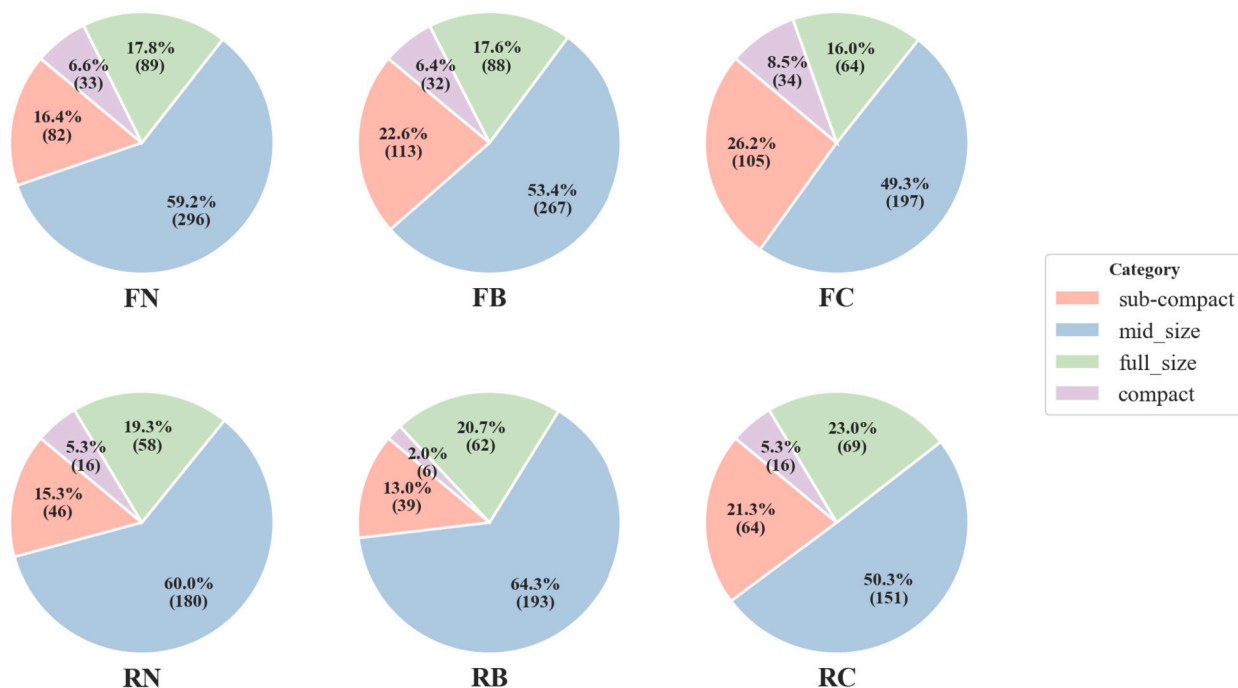


Fig. 2. Distribution of diversification of car model by size for each label. It is a pie chart showing how much distribution is indicated for each label by classifying the vehicle sizes into four. The abbreviations for each pie chart are as follows; FN: Front Normal, FB: Front Breakage, FC: Front Crushed, RN: Rear Normal, RB: Rear Breakage, RC: Rear Crushed.

Table 2
Configuration and Statistics of TQVCD dataset.

View	Type	Quantity
3/4 Front	Undamaged	500
	Breakage	500
	Undamaged	500
	Crushed	400
3/4 Rear	Undamaged	300
	Breakage	300
	Undamaged	300
	Crushed	300

3.2. Data preparation

Training neural networks with real-world raw images requires additional challenges, particularly concerning illumination, occlusion overlap, and other complexities [32]. RGB data, which includes color information, provides the advantages of capturing richer visual details and improving discrimination, suggesting valuable tasks such as recognizing different vehicle colors and assessing damage severity.

One significant challenge arises from illumination variations, present in real-world images, stemming from factors like changing weather conditions or inconsistent lighting sources [13]. These variations can obscure or exaggerate damage, impacting the model's ability to accurately classify it. Additionally, the presence of complex and cluttered backgrounds further complicates the task, as irrelevant elements in the image may divert the model's focus and lead to misclassification [25]. Addressing these challenges is crucial for ensuring the model's robustness in real-world scenarios and enhancing its accuracy in classifying vehicle damage.

On the contrary, employing grayscale data offers the potential to simplify the model architecture, reduce computational demands, and enhance robustness against noise and illumination variations. Grayscale data is often less sensitive to fluctuations in illumination and background noise compared to RGB data [2]. This reduction in sensitivity can contribute to improved model stability and performance in challenging real-world conditions.

However, it is important to acknowledge that grayscale images lack color information, which may limit their applicability in tasks that heavily rely on color cues [41]. For instance, tasks involving the identification of paint scratches or distinguishing between rust and dirt on a vehicle's surface may be compromised without color information [26].

Achieving a balance between information richness and computational efficiency becomes crucial in developing a classification system that is both robust and accurate in real-world scenarios. This balance ensures that the model can effectively leverage the advantages of grayscale data while still meeting the requirements of tasks dependent on color-specific details [6].

Considering the challenges and advantages associated with color information and computational efficiency, we have prepared two types of datasets: RGB and grayscale. In the domain of vehicle damage classification, the use of RGB data is intended to capture rich color information [36], thereby enhancing visual details and enabling precise discrimination of damage types based on color cues. RGB data proves valuable for tasks where color plays a crucial role in the classification process.

Conversely, the utilization of grayscale data serves the purpose of reducing computational demands, enhancing noise robustness under varying lighting conditions, and simplifying model architectures. This makes grayscale data suitable for scenarios where computational efficiency and robustness to illumination variations are of paramount importance [12]. By providing both RGB and grayscale datasets, we aim to offer flexibility and adaptability to different aspects of the vehicle damage classification task, ensuring the applicability of our approach across a range of real-world conditions.

4. Methodology

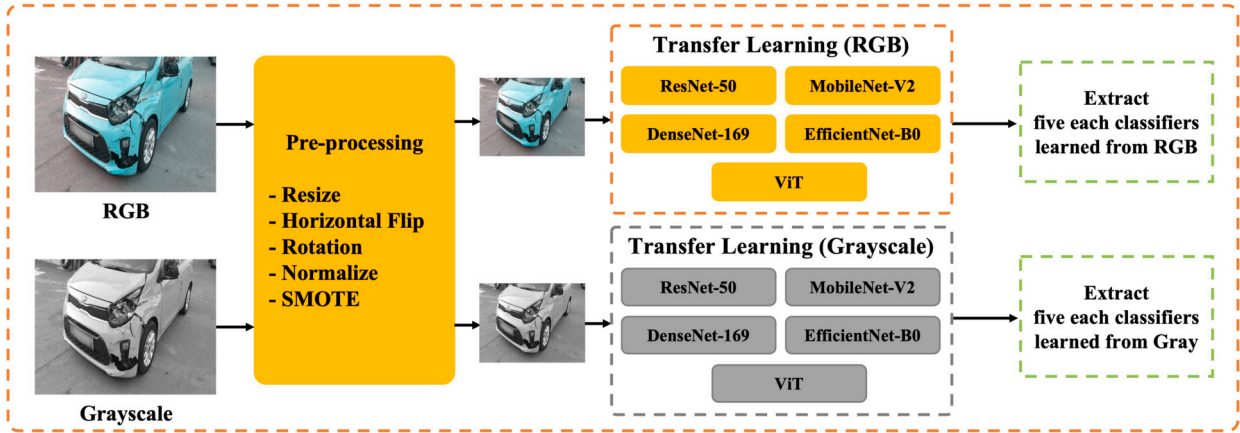
In this section, we introduce the sequence of our proposed pipeline. The pipeline is organized by two main components, data pre-processing and model implementation. First, we conduct the pre-processing steps applied to the data before feeding it into the model. These steps are essential for preparing the data and ensuring that it is in a suitable format for the subsequent stages of the pipeline.

Second, we describe the choice of models employed as the backbone network for our framework. The backbone network is one of the crucial elements, which contributes to the overall model performance in the vehicle damage classification. In addition, we specify the metrics employed to evaluate the model performance, providing a comprehensive assessment of their effectiveness. For more detailed procedures, the following subsections are examined with the overview of our framework (Fig. 3).

4.1. Proposed framework

Before model training, effective preprocessing is crucial for ensuring uniform and smooth training. Initially, we resize the input images to 224 x 224, facilitating collective data input for all models, considering the varying sizes of each dataset. Subsequently, data augmentation is applied to enhance data diversity, prevent overfitting, address class imbalances, and improve model robustness.

Phase 1: Training step - using transfer learning for single models for each breakage and crushed classifier.



Phase 2 : Inference step - Predict using model ensemble method.

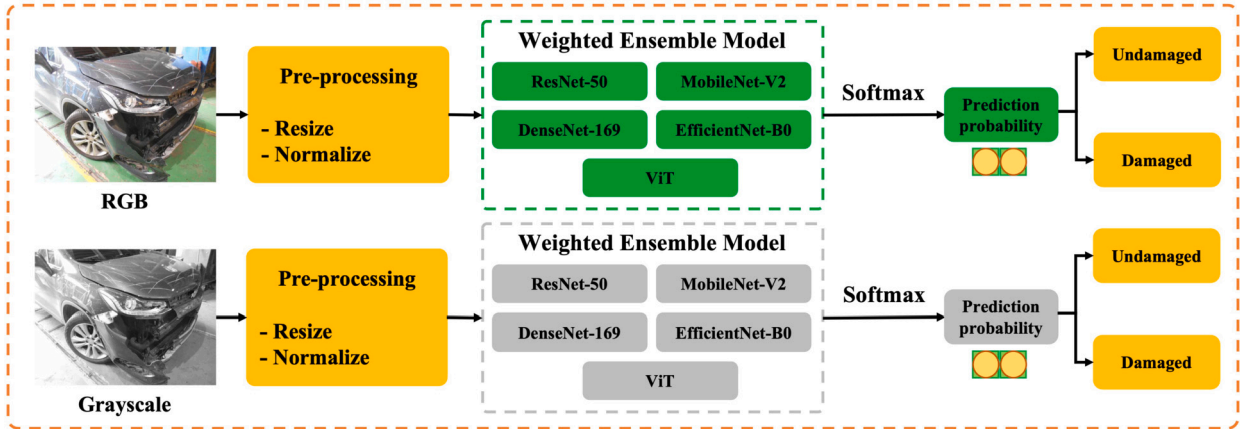


Fig. 3. The framework of our proposed vehicle damage classification classifier. In phase 1, we train the model using transfer learning techniques on RGB and Grayscale datasets and five deep learning architecture models, respectively. In the data preprocessing process, horizontal flip, rotation, and SMOTE techniques are used to improve the generalization ability of the model. In phase 2, we derive the final result using weighted soft voting, which optimizes the model's performance by weighting the predictions of each model. The binary classification task distinguishes between two categories: undamaged vs. breakage, and undamaged vs. crushed.

It enables models to perform well in diverse scenarios, proves particularly beneficial for limited data, and reduces training time by generating additional data through transformations without duplicating the original data.

Taking into account the characteristics of vehicle images, random horizontal flips are applied to robustly learn features in both left and right directions, leveraging the symmetric characteristics of vehicles. Additionally, random degree rotation is applied to accommodate potential variations in the angle of each image, considering the differing orientations present in vehicle images. To tackle an imbalanced dataset, we employ SMOTE (Synthetic Minority Over-sampling Technique), a method that generates synthetic instances for the minority class by interpolating between existing minority class samples [5]. This technique aids in improving classification performance by balancing class distribution and enhancing the representation of the minority class, thereby mitigating the impact of class imbalance on predictive modeling. We train each classifier for each damage type and vehicle orientation (front and rear) using a pre-trained deep-learning model on the TQVCD dataset. The pre-trained model, initially trained on the ImageNet dataset, is employed to learn the fundamental features of vehicles. By further training on the dataset, the model adapts its knowledge to focus on learning features specific to vehicle damage images, leveraging transfer learning.

To consolidate predictions, we employ a model ensemble technique through weighted soft voting on the individual models. This ensemble process enhances the robustness of the predictions, providing a more reliable outcome. Importantly, this process is performed separately on RGB datasets and grayscale datasets. RGB datasets, with their color information, can infer diverse details and differences, while grayscale datasets offer robustness against noise and illumination variations.

For more details on the transfer learning and backbone network utilized in our framework, Sections 4.2, 4.3, and 4.4 present evaluation indicators designed to assess the model performance, presenting a comprehensive understanding of its effectiveness in vehicle damage classification.

4.2. Transfer learning

Transfer learning is one of the highly effective approaches that leverage prior knowledge to expedite and refine similar tasks [42]. Its value is particularly pronounced when working with small and task-specific datasets, as pre-trained models can efficiently extract crucial image features while mitigating the overfitting risk. We utilize five widely recognized and transferability-verified pre-trained backbone models, while the details of the models are presented in Section 4.3.

These pre-trained models allow us to extract features and apply the learned weights to the specific task of vehicle damage classification. This approach differs from traditional machine learning methods, which necessitate beginning from scratch and learning individual tasks independently. Transfer learning enables us to draw upon informative features and knowledge from source tasks and efficiently apply them to the target task. In our context, the pre-trained model's classes serve as the source domain, while the target tasks involve the detection of various types of damages in the target domain. The success of knowledge transfer is particularly pronounced when there are similarities between the source and target domains, significantly enhancing the performance of the target task.

4.3. Backbone networks

We utilize several neural network architectures in computer vision applications, particularly for image classification tasks. We use Residual Network (ResNet) [37], Densely Connected Convolutional Network (DenseNet) [17], MobileNet [16], EfficientNet [35] and Vision Transformers (ViT) [10]. The following section provides a brief introduction to the backbone networks utilized.

4.3.1. ResNet-50

ResNet is a significant CNN model across various computer vision applications [37]. One of the most fundamental components in this model is the residual block. The concept of the residual involves the difference between the input data and the output of a layer. This residual is learned to approximate the identity mapping when the layer does not undergo any intended transformation. A key attribute of the residual block is the skip connection, which directly adds the input data to the output. This mechanism enables the residual to propagate through the network. Through the use of residual blocks, ResNet adeptly addresses the vanishing gradient problem, enabling the stacking of much deeper layers to extract rich features from images. In our study, we employ ResNet50, which encompasses 50 layers, showcasing its effectiveness in capturing intricate image features.

4.3.2. DenseNet-169

DenseNet, a CNN-based method recognized for its dense connectivity structure and effective feature recycling, has emerged as a prominent architecture in deep learning [17]. Unlike several traditional networks where each layer exclusively connects with its immediate successor, DenseNet adopts densely linked blocks, improving the exchange of information across layers. Each layer in DenseNet receives feature maps from all preceding layers, facilitating extensive gradient propagation and enabling the model to assimilate intricate representations. This architecture addresses the vanishing gradient problem, promoting the reuse of features and thereby enhancing training efficiency and overall performance. Various iterations of the DenseNet architecture, such as DenseNet-121, DenseNet-169, and DenseNet-201, offer different depths while adhering to the fundamental concept of dense connectivity. In our study, we conduct experiments using DenseNet-169, chosen for its balanced parameter number and computational efficiency among the variations of DenseNet.

4.3.3. MobileNet-V2

MobileNet, which has a CNN-based architecture, is specifically crafted to meet the efficiency and lightweight requirements of deep learning applications, particularly for devices with limited resources, such as embedded systems and mobile phones [16]. One of the main features is the utilization of depthwise separable convolutions, a technique that divides standard convolutions into separate layers for depthwise and pointwise convolutions. This division significantly reduces computational demands and model size. It presents some design variations such as MobileNet-V2 [30] and MobileNet-V3 [15], which further improve its performance by incorporating innovative concepts like inverted residuals, squeeze-and-excitation modules, and enhanced network design strategies. In our research, we conduct our experiments using MobileNet-V2, selected for its balance between efficiency and performance.

4.3.4. EfficientNet-B0

Based on a CNN architecture, EfficientNet is strategically designed to strike an optimal balance between model performance and computational efficiency [35]. It introduces the innovative concept of compound scaling, where the depth, width, and resolution of the model are uniformly scaled simultaneously. This approach yields models that excel in both accuracy and computational efficiency across a spectrum of tasks. Its scaling coefficients are determined via an automated process, which takes into account both accuracy and computational cost. It results in a family of models, ranging from EfficientNet-B0 to higher variants like EfficientNet-B7. We use EfficientNet-B0, emphasizing its lightweight design and computational efficiency.

4.3.5. ViT

Vision Transformers, called ViTs, represent a groundbreaking deep learning architecture, which harnesses the transformer framework initially developed for natural language processing and applies it to computer vision applications [10]. The fundamental concept

Table 3
Hyperparameter spaces of each model. The final hyperparameter is shown in bold.

Hyperparameter	Consideration
Train : Validation : Test	6:2:2
Learning rate	[1e-2, 3e-2, 3e-3, 1e-4, 3e-4]
Weight decay (fixed)	5e-4
Batch size	[16, 32 , 64, 128]
K-fold (fixed)	5
Early stopping patience (fixed)	5

involves dividing input images into fixed segments, or patches, embedding these patches along with positional embeddings, and subjecting them to transformer layers. By replacing traditional convolutional layers with self-attention mechanisms, Vision Transformers capture more contextual cues and demonstrate superiority over conventional convolutional neural networks across diverse datasets. This approach has shown exceptional effectiveness, particularly in tasks like image classification. We employ the ViT base model with a patch size of 16 and use an input resolution of 224 x 224.

4.4. Evaluation metrics

We evaluate our proposed model with four evaluation metrics: *Accuracy*, *Precision*, *Recall*, and *F1-score* [14]. These metrics are derived from the confusion matrix for describing the classification performance. The confusion matrix is organized by true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Each metric is computed as follows:

Accuracy is “a fundamental metric used to evaluate the performance of a classification model” [14]. It represents the percentage of correctly classified instances out of the total number of instances in the dataset. In other words, *accuracy* measures how well the model predicts both TP and TN. It provides an indication of the overall correctness of the model’s predictions across all classes. The equation is presented as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision is “a metric that evaluates the model’s ability to correctly identify positive instances (TP) among all the instances it has classified as positive (TP + FP)” [14]. In other words, *precision* assesses the accuracy of the positive predictions made by the model. It is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall measures “the model’s ability to identify all positive instances in the dataset” [14]. It evaluates the model’s capability to capture TP while minimizing FN. In other words, *recall* assesses the completeness of the model’s positive predictions. It is calculated as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

F1-score is “a metric that serves as the harmonic mean of precision and recall, offering a balanced performance assessment for classification tasks” [14]. By considering both *precision* and *recall*, *F1-score* provides a single score that reflects the model’s overall effectiveness. The formula for calculating the F1 score is as follows:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5. Experimental setup

We describe how to set up the experimental environment. All the experiments are performed with 12 GB RAM and 16 GB Tesla V100 GPU. The careful selection of appropriate hyperparameters is important in deep learning as it significantly influences the performance of a model [23]. We split the TQVCD dataset into train, validation, and test sets with a 6:2:2 ratio, employing a random seed. To determine optimal hyperparameters (e.g. learning rate and batch size), we utilize grid search, setting the values at 3e-4 and 32, respectively. The weight decay value is established as 5e-4 based on guidance from a relevant paper [19], which investigates weight decay values in relation to batch sizes.

We employ five-fold cross-validation and early stopping with a patience of 5. The hyperparameters space we explore is outlined in Table 3. We employ the binary cross-entropy loss function to effectively measure dissimilarity between predicted probabilities and actual binary labels [18]. The Adam optimizer is chosen for its effectiveness in training deep learning models [20].

6. Experimental results

We assess the effectiveness of the label space in the TQVCD dataset for classification using various neural network methods. We demonstrate our approach to ensure the robustness of model predictions through the use of ensemble methods. Model ensembles

Table 4

Performance of single model classifier on RGB dataset. B: Breakage classifier, C: Crushed classifier. On the RGB dataset, individual models' performance differed for each orientation and type of damage.

Model	Three-quarter view of front				Three-quarter view of rear			
	Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
B	ResNet-50	86.12	86.88	86.12	86.05	73.91	74.23	73.91
	DenseNet-169	96.17	96.44	96.17	96.17	95.65	95.65	95.65
	EfficientNet-B0	62.20	62.75	62.20	61.83	56.52	58.56	56.52
	MobileNet-V2	77.03	77.19	77.01	76.99	63.48	66.52	63.48
	ViT	97.61	97.72	97.61	97.61	62.61	63.23	62.61
C	ResNet-50	85.45	85.42	85.45	85.39	98.39	98.39	98.39
	DenseNet-169	97.58	97.59	97.58	97.57	89.52	89.55	89.52
	EfficientNet-B0	73.94	74.10	73.94	73.17	59.68	63.39	59.68
	MobileNet-V2	75.15	75.42	75.15	74.42	70.97	71.47	70.97
	ViT	79.39	79.42	79.39	79.10	54.84	58.03	54.84

involve combining predictions from multiple models to enhance overall performance and reliability. Then, we present an analysis demonstrating the robust and accurate predictions made by our proposed model on the test data. This section aims to showcase the model's effectiveness and reliability in making predictions across the diverse and challenging scenarios presented in the test dataset.

Table 4 presents the results for each single classifier applied to the RGB dataset. The ViT exhibited the highest performance for the breakage classifier with an accuracy of 97.61% in the three-quarter view of the front. This can be attributed to ViT's ability to effectively capture global features across complex and varied front-view images, such as the grille, headlights, and bumper. In contrast, the DenseNet-169 achieved the highest accuracy of 95.65% in the three-quarter view of the rear. The dense connectivity of DenseNet-169 promotes robust feature learning and gradient flow, making it particularly effective in learning less complex, homogeneous rear-view features.

For the crushed classifier, DenseNet-169 and ResNet-50 showed outstanding performance with accuracies of 97.58% and 98.39% in the front and rear views, respectively. DenseNet-169's dense connections facilitated effective feature reuse and learning, while ResNet-50's residual connections allowed for deep feature extraction, which is crucial for identifying subtle deformations in crushed parts.

The significant difference in ViT's performance between the breakage and crushed classifiers in the three-quarter view of the front can be explained by the distinct characteristics of the damage type. Breakages present clearer, more defined features that align well with ViT's strengths in global feature extraction, while crushed parts involve more complex, subtle deformations and potential visual clutter, making them harder for the model to classify accurately. Understanding the features of these dataset classes is essential for improving model performance and developing more effective damage classification systems.

The varying performance of models for the front and rear views in the RGB dataset highlights the importance of actively addressing dataset imbalance, data quality, and the distinctive features of damage types within each dataset.

To assess the effect of gloss on light reflection and noise among the different characteristics of the vehicle data, we provide the experimental results on the grayscale dataset in Table 5. The results show that DenseNet-169 performs best overall in the grayscale dataset, achieving 100% accuracy for the breakage classifier in the three-quarter view of the front and 93.97% in the rear view. For the crushed classifier, DenseNet-169 also shows the highest performance with 90.12% accuracy in the front view and 92.74% in the rear view.

DenseNet-169's higher performance in grayscale images is due to its architecture that promotes feature reuse through dense connectivity, effectively leveraging texture and shape features in the absence of color information. This allows for focused learning of fine-grained details, such as edges and textures. Additionally, grayscale images are less sensitive to variations in lighting and shadows, leading to more stable feature learning and higher classification accuracy, which is reflected in DenseNet-169's results.

The performance of all models improved for breakage in the three-quarter view of the front compared to the RGB dataset, demonstrating the significant impact of lighting on vehicle datasets. The lower classification performance for other classes compared to the breakage in the three-quarter view of the front indicates the need for additional data collection. To mitigate the performance variance of individual models within the scope of this dataset, we employ an ensemble approach.

The ensemble method utilizes soft voting, where label predictions are determined by averaging the predictions of individual models. Table 6 showcases the results of model ensembles for the RGB dataset.

The implementation of the weighted soft voting ensemble (W-SVE) method on the RGB dataset significantly improved model performance across all damage classifications compared to the soft voting ensemble (SVE) method. For the breakage classifier in the three-quarter view of the front, W-SVE achieved an accuracy of 98.09%, surpassing the SVE accuracy of 95.69%. This improvement highlights the effectiveness of assigning appropriate weights to individual model predictions, leading to more accurate and robust results. Similarly, for the rear view, W-SVE improved the accuracy from 79.13% (SVE) to 91.30%, demonstrating a substantial enhancement in performance.

For the crushed classifier, the benefits of W-SVE were also evident. In the three-quarter view of the front, W-SVE increased the accuracy from 89.70% to 95.15%. In the rear view, the accuracy improved from 91.94% to 95.97%, indicating that the weighted approach effectively mitigates the performance variance and leverages the strengths of each model to achieve higher overall accuracy and reliability.

Table 5

Performance of single model classifier on the grayscale dataset. B: Breakage classifier, C: Crushed classifier. DenseNet-169 performed best on the grayscale dataset in both orientations and damage types.

Model		Three-quarter view of front				Three-quarter view of rear			
		Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
B	ResNet-50	97.46	97.46	97.46	97.46	81.70	82.61	80.17	80.39
	DenseNet-169	100.00	100.00	100.00	100.00	93.97	94.00	93.97	93.98
	EfficientNet-B0	99.49	99.50	99.49	99.49	62.93	72.77	62.93	62.11
	MobileNet-V2	99.49	99.50	99.49	99.49	64.46	65.79	62.93	63.31
	VIT	97.46	97.57	97.46	97.45	64.46	63.50	64.66	62.17
C	ResNet-50	82.72	85.35	82.72	81.88	66.94	68.99	66.94	66.97
	DenseNet-169	90.12	90.92	90.12	90.18	92.74	92.77	92.74	92.75
	EfficientNet-B0	70.99	71.01	70.99	70.16	50.00	54.71	50.00	47.60
	MobileNet-V2	69.75	69.48	69.75	69.40	70.16	70.60	70.16	70.27
	VIT	76.54	76.54	76.54	76.54	58.87	58.03	58.87	54.85

Table 6

Performance of ensemble model classifier on RGB dataset. B: Breakage classifier, C: Crushed classifier, SVE: Soft voting ensemble, and W-SVE: Weighted soft voting ensemble. We conducted an ensemble on the RGB dataset to ensure the robustness of individual model performances. Our approach employed a weighted soft voting ensemble, carefully considering the weight of each model, resulting in a demonstration of robust performance.

	Model	Three-quarter view of front				Three-quarter view of rear			
		Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
B	SVE	95.69	95.73	95.69	95.69	79.13	79.41	79.13	79.17
	W-SVE	98.09	98.16	98.09	98.09	91.30	91.33	91.30	91.20
C	SVE	89.70	89.74	89.70	89.63	91.94	92.18	91.94	91.96
	W-SVE	95.15	95.22	95.15	95.13	95.97	95.97	95.97	95.96

Table 7

Performance of ensemble model classifier on the grayscale dataset. B: Breakage classifier, C: Crushed classifier, SVE: Soft voting ensemble, and W-SVE: Weighted soft voting ensemble. We conducted an ensemble on the grayscale dataset to improve the robustness of individual model performances, showcasing robust performance through the utilization of a weighted soft voting ensemble that considered the weight of each model.

	Model	Three-quarter view of front				Three-quarter view of rear			
		Accuracy (%)	Precision	Recall	F1 Score	Accuracy (%)	Precision	Recall	F1 Score
B	SVE	100.00	100.00	100.00	100.00	82.76	83.88	82.76	82.93
	W-SVE	100.00	100.00	100.00	100.00	93.10	93.21	93.10	93.13
C	SVE	83.33	83.64	83.33	83.08	79.84	80.51	79.84	79.92
	W-SVE	91.36	91.35	91.36	91.34	89.52	89.68	89.52	89.54

Table 7 shows the results of model ensembles for the grayscale dataset. On the grayscale dataset, the weighted soft voting ensemble (W-SVE) method yielded remarkable improvements in model performance compared to the soft voting ensemble (SVE) method. For the breakage classifier in the three-quarter view of the front, both SVE and W-SVE achieved perfect accuracy (100.00%). However, for the rear view, W-SVE significantly enhanced the accuracy from 82.76% (SVE) to 93.10%. This result underscores the ability of W-SVE to maintain high performance even under varying conditions by appropriately weighting the contributions of individual models.

The crushed classifier also benefited from the W-SVE approach. In the three-quarter view of the front, W-SVE improved the accuracy from 83.33% (SVE) to 91.36%. For the rear view, the accuracy experienced a notable increase from 79.84% to 89.52%. These results illustrate that the weighted soft voting ensemble not only enhances the robustness of model predictions but also ensures more consistent and higher classification accuracy across different views and damage types.

The application of the weighted soft voting ensemble method has proven to improve robust model performance on both RGB and grayscale datasets. By carefully considering and assigning weights to each model, W-SVE mitigates individual model performance variances and leverages their collective strengths, resulting in more accurate and reliable damage classification. These findings demonstrate the efficacy of the weighted ensemble approach in achieving robust performance across diverse conditions and damage types, making it a valuable technique for vehicle damage classification systems.

6.1. Interview results

Given the industrial relevance of the topic, it is essential to integrate insights from both industry professionals and academic researchers to bolster the foundational aspects of this paper. To validate the practical effectiveness of this study, we consulted three

industry experts, one of whom is from the used-car industry, which is among the many industries that manage large vehicles. Below is a summary of their responses to the questions.

• **Q1. How would the integration of an AI-based vehicle damage classification model impact the used-car industry?**

All respondents answered positively to Q1. While vehicle insurance companies and large companies operating in the used-car industry may already be utilizing AI technology, most small and medium-sized companies still rely on physical inspections, often conducting on-site visits and assessing vehicles through phone calls and image transmission prior to purchase. Respondents expressed that the introduction of an AI-based classification model for identifying the extent of vehicle damage would reduce workload and have a positive impact by enhancing the company's competitiveness.

• **Q2. When considering its applicability to real-world industries, which dataset do you find more beneficial: the fine-grained dataset or the TQVCD dataset?**

Regarding Q2, all respondents expressed positive opinions regarding the TQVCD dataset. When consumers sell a used car, dealers typically receive a video showcasing the entire vehicle. Similar to the TQVCD dataset, these videos capture the entire exterior of the vehicle. Although the videos may contain segmented classifications, humans still need to visually inspect the extent of damage. Before personal inspection, a comprehensive understanding of the vehicle's general condition is necessary. Respondents indicated that the absence of segmented classification by parts was not significant, as the three-sided video provided sufficient information about the parts requiring replacement or repair. Therefore, they suggested that it is more meaningful to establish a three-quarters view dataset that allows for the assessment of the entire vehicle, rather than fine-grained data organized by parts.

• **Q3. What do you think about dividing the damage type into breakage and crushed?**

Regarding Q3, all respondents provided negative feedback. In the used-car industry, vehicles with severe damage, as indicated by the TQVCD dataset, are often not purchased. Instead, there is a demand for more research on assessing minor damages such as dents, cracks, separations, or scratches. When purchasing a used car, setting a specific price without an in-person inspection, especially regarding exterior damage, is challenging. For instance, assessing exterior scratches requires consideration of various factors such as the number, length, and depth of the scratches as a whole. Particularly, depth is considered a more critical factor than length. Deep scratches may lead to frame corrosion, significantly impacting depreciation. While surface scratches mainly cause cosmetic issues, deep scratches expose the metal to moisture and chemicals, increasing the risk of corrosion. This factor greatly influences the depreciation of used cars and the decision-making process for purchasing them.

7. Conclusion and future work

For the large-scale vehicle management sector (e.g., used-car platforms, car-sharing platforms, and vehicle insurance companies), there is a compelling need for an automated vehicle damage classification process. However, several challenges arise when dealing with vehicle damage classification tasks using real-world data. First, numerous critical issues emerge concerning data quality, labeling complexity, and the overall need for model reliability. Second, the lack of public data relative to the industry scale that manages large vehicles limits active research in this area.

To address these challenges, we focused on reducing the complexity of labeling data by vehicle parts and introduced a novel perspective: the Three-Quarter View Car Damage Dataset (TQVCD dataset). We constructed the TQVCD dataset based on the orientation and type of damage to the vehicle while maintaining a three-quarter view. The classes in the TQVCD dataset consisted of "broken," "crushed," and "undamaged."

We extensively validated the reliability of the TQVCD dataset through the following process. First, we prepared the data in both RGB and grayscale during the data preparation process. While RGB data captures richer visual details and improves discrimination, it suffers from inconsistent illumination. To address this issue, we also prepared a grayscale dataset.

We utilized five widely adopted pre-trained deep learning architectures (ResNet-50, DenseNet-169, EfficientNet-B0, MobileNet-V2, and ViT) commonly used in computer vision tasks. Transfer learning techniques were applied to train models on the TQVCD dataset. We evaluated the performance of each model classifier across various categories of vehicle damage. When applied to RGB datasets, different models showed varying performance levels depending on the type of vehicle damage and its orientation. However, DenseNet-169 consistently outperformed other models on grayscale datasets across all scenarios.

To enhance prediction reliability, we implemented a weighted soft voting approach, where multiple deep learning models contribute to the prediction by considering the weighted average of the prediction probabilities. This ensemble method aims to integrate the strengths of each model while addressing potential weaknesses.

The results demonstrated that the ensemble model is robust against the performance of individual models on both RGB and grayscale datasets. These experiments were crucial for ensuring that the models are resilient and effective in diverse and challenging real-world scenarios, thereby contributing to the overall reliability of the proposed vehicle damage classification system.

To evaluate the effectiveness of this study from an industrial perspective, we interviewed three industry experts who work on a used-car platform that manages a large number of vehicles. They responded positively to the technology of managing damaged vehicles using AI, particularly showing enthusiasm for our TQVCD dataset, which determines the overall state of the vehicle using

a three-quarters view. However, they expressed reservations about dividing the types of damage into “breakage” and “crushed.” We found that minor damages (e.g., scratches, minor crushes, and minor breakages) need to be given more consideration when applied to the used-car industry.

However, our study has some limitations. Firstly, there is a noticeable data imbalance problem, resulting in significant differences in model performance. For example, the significant drop in the performance of crushed classification compared to breakage for the front view is due to the lack of data for crushed instances. Secondly, there is a data diversification imbalance for compact cars in the rearview compared to the front, necessitating additional data acquisition to enhance the completeness of the dataset. Thirdly, the continuous addition of data may lead to variations in the ensemble’s weight values. Therefore, proper consideration and adjustment of these weight values for each classifier are necessary when additional learning of the dataset occurs.

In our future research, we aim to address these limitations by exploring the vehicle’s side view and expanding our classification capabilities to include minor damages such as scratches. In addition, we plan to extend our approach to incorporate RGB-D data, introducing depth information to enhance the richness of details extracted from a single image. Ultimately, integrating these advancements, coupled with the inclusion of existing fine-grained datasets, is anticipated to significantly contribute to the robustness and effectiveness of automated vehicle classification tasks.

CRedit authorship contribution statement

Donggeun Lee: Writing – original draft, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Juyeob Lee:** Writing – review & editing, Validation, Software, Methodology, Investigation. **Eunil Park:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The TQVCD dataset is available at <https://github.com/dxlabskku/TQVCD.git>.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-00421, AI Graduate School Support Program (Sungkyunkwan University); No. IITP-2020-0-01816, ICAN (ICT Challenge and Advanced Network of HRD) program).

References

- [1] AI Hub, Ai hub, <https://aihub.or.kr>. (Accessed 21 August 2022), 2002.
- [2] A. Andreopoulos, J.K. Tsotsos, On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2011) 110–126.
- [3] C.T. Artan, T. Kaya, Car damage analysis for insurance market using convolutional neural networks, in: *Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making: Proceedings of the INFUS 2019 Conference*, Istanbul, Turkey, July 23–25, 2019, Springer, 2020, pp. 313–321.
- [4] B. Balci, Y. Artan, B. Alkan, A. Elihos, Front-view vehicle damage detection using roadway surveillance camera images, in: *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2019, pp. 193–198.
- [5] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *J. Artif. Intell. Res.* 16 (2002) 321–357.
- [6] H.D. Cheng, X.H. Jiang, Y. Sun, J. Wang, Color image segmentation: advances and prospects, *Pattern Recognit.* 34 (2001) 2259–2281.
- [7] A.C. Chua, C.R.B. Mercado, J.P.R. Pin, A.K.T. Tan, J.B.L. Tinhay, E.P. Dadios, R.K.C. Billones, Damage identification of selected car parts using image classification and deep learning, in: *2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, IEEE, 2021, pp. 1–5.
- [8] J. De Deijn, Automatic car damage recognition using convolutional neural networks, in: *2018 Internship Report MSc Business Analytics*, 2018.
- [9] N. Dhieb, H. Ghazzai, H. Besbes, Y. Massoud, A very deep transfer learning model for vehicle damage detection and localization, in: *Proc. of ICM '19, IEEE*, 2019, pp. 158–161.
- [10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, in: *Proceedings of ICLR '21*, 2021, pp. 1–21.
- [11] M. Dwivedi, H.S. Malik, S. Omkar, E.B. Monis, B. Khanna, S.R. Samal, A. Tiwari, A. Rath, Deep learning-based car damage classification and detection, in: *Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2019*, Springer, 2021, pp. 207–221.
- [12] H. Fitriyah, R.C. Wihandika, An analysis of rgb, hue and grayscale under various illuminations, in: *Proc. of SIET '18, IEEE*, 2018, pp. 38–41.
- [13] G. Fountas, A. Fonzone, N. Gharavi, T. Rye, The joint effect of weather and lighting conditions on injury severities of single-vehicle accidents, *Anal. Methods Accid. Res.* 27 (2020) 100124.
- [14] M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations, *Int. J. Data Min. Knowl. Manag. Process* 5 (2015) 1–11.
- [15] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: *Proceedings of ICCV '19, 2019*, pp. 1314–1324.
- [16] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, <https://arxiv.org/abs/1704.04861>, 2017.
- [17] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: *Proceedings of CVPR '17, 2017*, pp. 4700–4708.

- [18] H. Ji, C. An, M. Lee, J. Yang, E. Park, Fused deep neural networks for sustainable and computational management of heat-transfer pipeline diagnosis, *Dev. Built Environ.* 14 (2023) 100144.
- [19] N.S. Keskar, J. Nocedal, P.T.P. Tang, D. Mudigere, M. Smelyanskiy, On large-batch training for deep learning: generalization gap and sharp minima, in: *Proceedings of ICLR '17*, 2017, pp. 1–16.
- [20] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, <https://arxiv.org/abs/1412.6980>, 2014.
- [21] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: *Proc. of ICCV '13 Workshop*, 2013, pp. 554–561.
- [22] P.M. Kyu, K. Woraratpanya, Car damage detection and classification, in: *Proc. of IAIT '20*, 2020, pp. 1–6.
- [23] S. Lee, D. Jeong, E. Park, Multiemo: multi-task framework for emoji prediction, *Knowl.-Based Syst.* 242 (2022) 108437.
- [24] P. Li, B. Shen, W. Dong, An anti-fraud system for car insurance claim based on visual evidence, <https://arxiv.org/abs/1804.11207>, 2018.
- [25] Z. Liu, Z. Li, L. Li, H. Yang, Complex background classification network: a deep learning method for urban images classification, *Comput. Electr. Eng.* 87 (2020) 106771.
- [26] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* 9 (1979) 62–66.
- [27] N. Patel, S. Shinde, F. Poly, Automated damage detection in operational vehicles using mask r-cnn, in: *Advanced Computing Technologies and Applications: Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA 2020*, Springer, 2020, pp. 563–571.
- [28] K. Patil, M. Kulkarni, A. Sriraman, S. Karande, Deep learning based car damage classification, in: *Proc. of ICMLA '17*, IEEE, 2017, pp. 50–54.
- [29] J. Qaddour, S.A. Siddiqua, Automatic damaged vehicle estimator using enhanced deep learning algorithm, *Intell. Syst. Appl.* 18 (2023) 200192.
- [30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: *Proc. of CVPR '18*, 2018, pp. 4510–4520.
- [31] T. Seo, K.H. Park, H. Chung, Socar: socially-obtained car dataset for image recognition in the wild, in: *Proc. of WACV '23*, 2023, pp. 430–438.
- [32] L. Shao, Z. Cai, L. Liu, K. Lu, Performance evaluation of deep feature learning for rgb-d image/video classification, *Inf. Sci.* 385 (2017) 266–283.
- [33] R. Singh, M.P. Ayyar, T.V.S. Pavan, S. Gosain, R.R. Shah, Automating car insurance claims using deep learning techniques, in: *Proc. of BigMM '19*, IEEE, 2019, pp. 199–207.
- [34] C. Sruthy, S. Kunjumon, R. Nandakumar, Car damage identification and categorization using various transfer learning models, in: *Proc. of ICOEI '21*, IEEE, 2021, pp. 1097–1101.
- [35] M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks, in: *Proc. of ICML '19*, PMLR, 2019, pp. 6105–6114.
- [36] J. Tanaka, D. Weiskopf, P. Williams, The role of color in high-level vision, *Trends Cogn. Sci.* 5 (2001) 211–215.
- [37] S. Targ, D. Almeida, K. Lyman, Resnet in resnet: generalizing residual architectures, <https://arxiv.org/abs/1603.08029>, 2016.
- [38] X. Wang, W. Li, Z. Wu, Cardd: a new dataset for vision-based car damage detection, *IEEE Trans. Intell. Transp. Syst.* 24 (2023) 7202–7214.
- [39] U. Waqas, N. Akram, S. Kim, D. Lee, J. Jeon, Vehicle damage classification and fraudulent image detection including moiré effect using deep learning, in: *Proc. of CCECE '20*, IEEE, 2020, pp. 1–5.
- [40] L. Yang, P. Luo, C. Change Loy, X. Tang, A large-scale car dataset for fine-grained categorization and verification, in: *Proc. of CVPR '15*, 2015, pp. 3973–3981.
- [41] I. Žeger, S. Grgić, J. Vuković, G. Šišul, Grayscale image colorization methods: overview and evaluation, *IEEE Access* 9 (2021) 113326–113346.
- [42] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, *Proc. IEEE* 109 (2020) 43–76.