Construction and Building Materials 342 (2022) 127997

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Deep transfer learning-based vehicle classification by asphalt pavement vibration

Fangyu Liu [a], Zhoujing Ye [b], Linbing Wang [a,*]

[a] Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
[b] National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Keywords:
Deep transfer learning
CNN model
Asphalt pavement
Vibration signals
Vehicle classification

ABSTRACT

Deep transfer learning (TL) has great potential for a wide range of applications in civil engineering. This work aims to propose a deep transfer learning-based method for vehicle classification by asphalt pavement vibration. This work first used the pavement vibration IoT monitoring system to collect raw vibration signals and performed the wavelet transform to obtain denoised vibration signals. The vibration signals were then represented in two different ways, including the time-domain graph and the time-frequency graph. Finally, two deep transfer learning-based methods, namely Method I (Time-domain & TL) and Method II (Time-frequency & TL), were applied for vehicle classification according to the two different representations of vibration signals. The results show that the CNN model had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95. The CNN model in Method II performed better in the accuracy metrics with considering label imbalance, but worse in the accuracy metrics without considering label imbalance than that in Method I. The differences between these two methods have been investigated and discussed in detail in terms of input types, accuracy metrics, and application prospects. The CNN model with deep transfer learning could be an effective, accurate, and reliable technique for vehicle classification based on asphalt pavement vibration.

1. Introduction

How to effectively, rapidly, and accurately classify vehicles? Vehicle classification plays an important role in intelligent traffic management and pavement performance evaluation. Currently, various technologies are used for vehicle classification and they are divided into two main groups, non-intrusive and intrusive technologies. Non-intrusive technologies contain video imaging (e.g. cameras) and radar [1,2], while intrusive technologies include piezoelectric sensors, inductive loops, fiber optic sensors, and integrated electronic piezoelectric accelerometers [3–6]. Non-intrusive technologies are sensitive to traffic and weather conditions, while the monitoring information and accuracy are limited [7,8]. Intrusive technologies strongly relied on the sensor, resulting in high installation and maintenance costs [7,9]. The weigh-in-motion (WIM) systems are the typical and popular intrusive technologies, while they consist of various sensors and can collect numerous traffic information, such as vehicle types, vehicle speed, vehicle axle counts, and vehicle weight. Due to the high requirements for pavement stiffness and roughness, the pavement needs to be improved to install the WIM systems, increasing the construction and maintenance costs. Apart from high costs, the traditional WIM systems are often susceptible to traffic or weather disruptions [9]. In addition, those sensors used for intrusive technologies sometimes have to be equipped with additional adapters and data acquisition devices due to the lack of data processing, storage, and communication techniques [8,9]. These limit the widespread use of those sensors and make real-time monitoring more difficult.

To overcome the aforementioned limitations, an acceleration sensor based on the Micro-Electro-Mechanical System (MEMS) was developed and applied in the asphalt pavement vibration monitoring. MEMS vibration sensors show strong advantages in terms of low power consumption, miniaturization, and scalability, meanwhile they achieve wireless communication functions with the help of the Internet of Things (IoT) technology. Numerous researchers have successfully applied MEMS vibration sensors to collect traffic information. Bajwa et al. [7] developed a wireless sensor network for vehicle classification based on axle count and spacing and this network included vibration sensors (measure pavement acceleration), vehicle detection sensors (monitor

* Corresponding author.
E-mail addresses: lfangyu@vt.edu (F. Liu), yezhoujing@ustb.edu.cn (Z. Ye), wangl@vt.edu (L. Wang).

https://doi.org/10.1016/j.conbuildmat.2022.127997
Received 1 February 2022; Received in revised form 13 May 2022; Accepted 29 May 2022
Available online 9 June 2022

vehicle's time of arrival and departure), and an access point (record the sensors' data). Ma et al. [10] proposed a prototype axle count and spacing automatic vehicle classification system based on wireless accelerometer sensors (detect vehicle axles) and magnetometer sensors (estimate vehicle speed). Huang et al. [11] developed an IoT-based wireless sensor system (based entirely on wireless accelerometer sensors) for traffic volume and vehicle classification, while they conducted a series of laboratory tests, field tests, and numerical simulations to validate the monitoring system. Ye et al. [9] collected comprehensive traffic information by a vibration-based in-field pavement monitoring system (MEMS acceleration sensor) and applied the artificial neural network and the k-means++ cluster analysis for vehicle classification and the abnormal vehicle weight, separately. Duarte and Hu [12] investigated the wireless sensor network for vehicle classification, while they built a dataset extracted from this wireless sensor network and applied machine learning models for data conditioning and classification. Although all these works used wireless sensors, some of them installed the sensors on the roadside, which might lead to the missing the real pavement vibration caused by vehicle load. This is because pavement vibration fades rapidly during the transmission in the pavement and pavement vibration measured by the sensors on the roadside tends to be smaller than those caused by vehicle loads [13]. Therefore, the pavement vibration should be collected in the center of the road, where it is usually directly subjected to vehicle loads.

Another issue is how to accurately and efficiently classify vehicles by pavement vibration. The vehicle type and pavement vibration are more likely to have a nonlinear relationship and traffic-induced vibration is often influenced by numerous factors, such as temperature, vehicle speed, location of vehicle loads, pavement structure, pavement roughness, and so on [9,14]. It is quite difficult to propose an empirical and simple function to establish the relationship between vehicle type and pavement vibration. Deep learning is becoming an ideal solution. In recent decades, deep learning has been widely applied in civil engineering [15–23]. As showing the powerful non-linearity and automatic feature extraction capacities, deep learning, especially convolutional neural network (CNN), has become a popular and efficient tool for vibration signal processing. Yu et al. [15] proposed a CNN-based method to identify and localize the damage of building structures by detecting vibration signals from structures. Dang et al. [16] developed a practical end-to-end framework using a hybrid deep learning model that incorporated CNN and long-short term memory (LSTM), which they applied to process vibration signals and detect the damage. Pal et al. [17] developed a CNN-based method to process the vibration signals of a steel frame structure with bolted connections and then detect the damage condition. Alazzawi and Wang [18] proposed a new deep learning-based structural health monitoring method that combined the raw time-domain structural response signals and the deep residual network. Although these works achieved satisfactory accuracy, the size of their datasets tended to be relatively small. This limits the performance of deep learning models, especially CNN which typically requires a large dataset to maximize its performance. In this case, transfer learning (TL) attracts more attention as it relaxes the requirement for the large dataset. How can transfer learning achieve it? Transfer learning is defined as: given a source domain D_S and learning task T_S, and a target domain D_T and learning task T_T, transfer learning aims to improve the performance of predictive function for learning task T_T by transferring the knowledge from D_S and T_S, where $D_S \neq D_T$ and $T_S \neq T_T$, and the size of D_S is much larger than that of D_T in most cases [24,25]. When transfer learning is applied in deep neural networks, it is often called deep transfer learning which examines how to leverage knowledge from other domains through deep neural networks [25]. Deep transfer learning is divided into four categories, instance-based deep transfer learning, mapping-based deep transfer learning, network-based deep transfer learning, and adversarial-based deep transfer learning [25]. Deep transfer learning needs a small amount of labeled data in the target domain to fine-tune the pre-trained deep learning models that have already been trained by a large dataset in the source domain. Deep transfer learning has started to be applied in civil engineering. Gao and Mosalam [26] applied deep transfer learning and CNN to structural damage recognition with two strategies, namely feature extractor and fine-tuning. Gopalakrishnan et al. [27] employed deep transfer learning and CNN to automatically detect cracks in pavement images, which include a large number of defects and non-crack anomalies. Kim et al. [28] developed a CNN-based construction object-detection method to accurately identify construction equipment with the help of deep transfer learning. Li et al. [29] proposed a deep transfer learning-based framework for estimating missing sensor data for high arch dams, which was based on the deep-stacked bidirectional LSTM model. These demonstrate that deep transfer learning has great potential for a wide range of applications in civil engineering.

Therefore, this work aims to propose a deep transfer learning-based vehicle classification by asphalt pavement vibration. The key contributions of this work are summarized as follows:

- The wavelet transform was applied to denoise the pavement vibration signals which were collected by the pavement vibration IoT monitoring system.
- The pavement vibration signals were represented in two different ways, including the time-domain graph and the time-frequency graph.
- Two deep transfer learning-based methods were proposed for vehicle classification according to the two different representations of the pavement vibration signals.

2. Methods

Fig. 1 shows the framework of this work. It aims to apply deep transfer learning for vehicle classification by asphalt pavement vibration and includes three main steps. Firstly, a pavement vibration IoT monitoring system was used to monitor pavement vibration, and raw vibration signals were collected through this system. Secondly, raw vibration signals were denoised by the wavelet transform to obtain denoised vibration signals. Finally, deep transfer learning was applied for vehicle classification. During this step, two deep transfer learning-based methods, namely Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for different types of inputs (time-

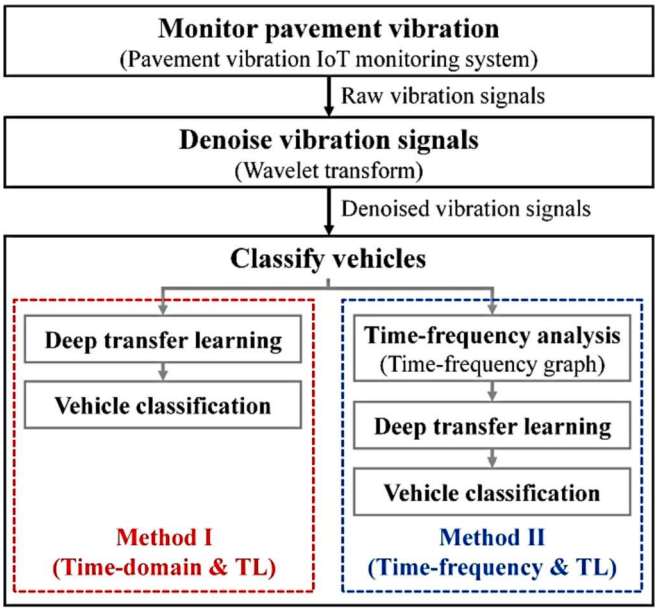

Fig. 1. Framework of deep transfer learning-based vehicle classification by asphalt pavement vibration.

domain graph and time-frequency graph). In Method I, the CNN model processed time-domain graphs to classify vehicles. In Method II, the CNN model employed time-frequency graphs, which were the results of time-frequency analysis, for vehicle classification.

2.1. Monitoring pavement vibration

A pavement vibration IoT monitoring system was used to efficiently collect pavement vibration in real-time [8]. This system consisted of acceleration sensing nodes, a gateway, and a cloud platform. This system has been applied in the field to monitor asphalt pavement vibration [8]. Fig. 2 shows the field layout of the pavement vibration IoT monitoring system. For each traffic direction, two rows of acceleration sensing nodes were deployed, with 7 acceleration sensing nodes in each row. The distance between the two acceleration sensing nodes was 50 cm and the distance between the acceleration sensing node and the roadside was 25 cm and 50 cm in two rows, respectively. The acceleration sensing nodes applied the MEMS technology to support the integration of MEMS sensors, processors, and other electronic components and the detailed information on acceleration sensing nodes could be found in the literature [8]. The acceleration sensing nodes collected pavement vibration signals at a sampling frequency of 500 Hz. Based on low power consumption and high scalability, this system enabled the monitoring, transmission and analysis of pavement vibration signals [8]. More details could be found in the literature [8].

2.2. Denoising vibration signals

2.2.1. Wavelet transform

The raw vibration signals collected by the pavement vibration IoT monitoring system contained noises that might affect the accuracy of vehicle classification, highlighting the importance of denoising. Wavelet transform could decompose a signal into several scales representing different frequency bands, and at each scale, the location of the instantaneous structure of the signal can be approximated [30]. Based on this property, wavelet transform has become a popular method for denoising vibration signals [30] in civil engineering [31,32]. Compared with continuous wavelet transform (CWT), discrete wavelet transform (DWT) is more suitable for denoising signals and more computationally efficient [32]. The threshold function and the wavelet function (i.e. the mother wavelet) play a crucial role in the signal denoising by DWT. The threshold function includes the hard thresholding, the soft thresholding, the compromising method of the hard and soft thresholding, and others. The compromising method of the hard and soft thresholding was proposed in the literature [33] and in the compromising method, the threshold values of the detail coefficients are defined as:

$$\widehat{\omega}_{j,k} = \begin{cases} sign(\omega_{j,k})\left(|\omega_{j,k}| - \alpha\lambda\right), & |\omega_{j,k}| \geq \lambda \\ 0, & |\omega_{j,k}| < \lambda \end{cases}, (0 \leq \alpha \leq 1) \quad (1)$$

where, $\widehat{\omega}_{j,k}$ is the estimated wavelet coefficients, $\omega_{j,k}$ is the wavelet coefficients, λ is the selected threshold, *sign*() is the sign function, α is usually 0.5.

To determine the threshold, Johnstone and Silverman [34] defined a level-dependent threshold applied at each detail level for processing various noises and the threshold is defined as follows:

$$\lambda_m = \sigma_m \times \sqrt{2log(N)} = MAD_m \Big/ 0.6745 \times \sqrt{2log(N)} \quad (2)$$

where MAD_m is the median absolute value of the detail coefficients at level m and N is the length of the signal.

The wavelet function is critical in practice as it affects the performance of DWT [32]. Several wavelet functions were investigated in this paper, including Biorthogonal (bior), Symlets (sym), Coiflets (coif), Reverse biorthogonal (rbio), and Daubechies (db), while some wavelet functions are presented in Fig. 3. The wavelet functions used in this work are based on a Python library named 'PyWavelets' [35].

2.2.2. Evaluation metrics

Various quantitative evaluation metrics have been used to evaluate the performance of signal denoising in terms of the reconstructed signal quality. The common evaluation metrics include the signal-to-noise ratio (SNR) and the root mean square error (RMSE). The SNR and RMSE were defined as follows [32,36].

- SNR:

$$SNR = 10log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \quad (3)$$

- RMSE:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \quad (4)$$

Where y_i is the original signal and y_i' is the denoised signal.

2.3. Classifying vehicles

During the process of classifying vehicles, two methods were proposed based on deep transfer learning (Fig. 1). The difference between these two methods lies in the input image. Method I (Time-domain & TL)

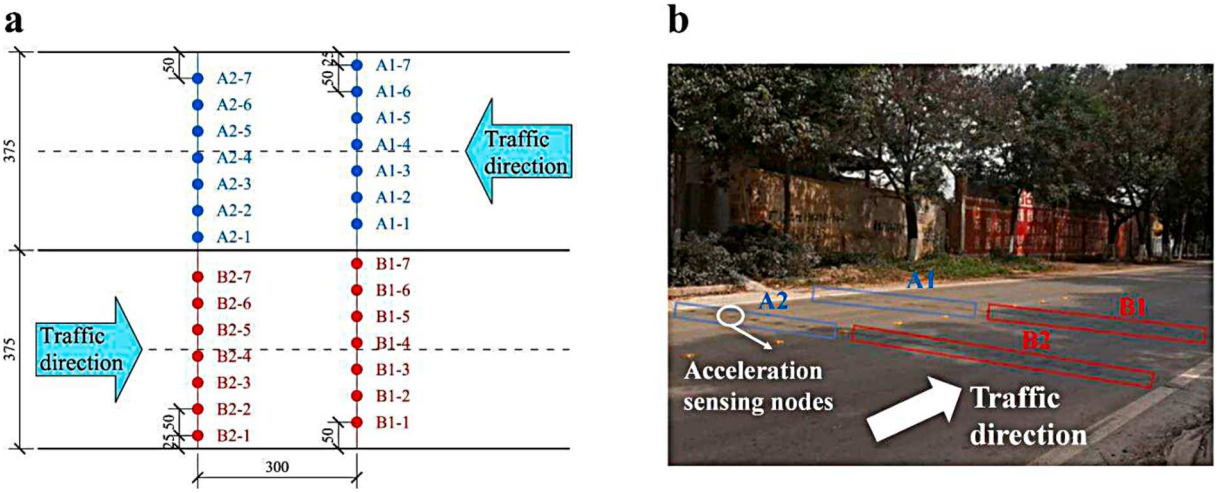

Fig. 2. The field layout of the pavement vibration IoT monitoring system. (a) Scheme and (b) image of the on-site monitoring.

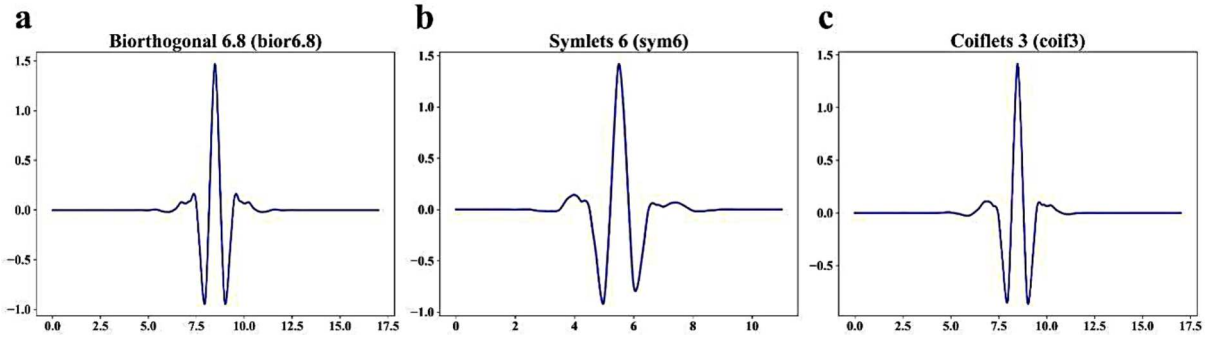

Fig. 3. Wavelet functions used in this work. (a) Biorthogonal 6.8 (bior6.8), (b) Symlets 6 (sym6), and (c) Coiflets 3 (coif3).

was proposed based on the time-domain graph and Method II (Time-frequency & TL) was developed based on the time-frequency graph which was the result of time-frequency analysis.

2.3.1. Time-domain graph and Time-frequency graph

The vibration signal has both time and frequency features, and accordingly, the vibration signal is represented as a function of time and frequency. Based on time-domain and frequency-domain representations, the signal graphs are divided into the time-domain graph and the frequency-domain graph. In short, the time-domain graph shows how the signal varies over time (Fig. 4a), while the frequency-domain graph shows how many signals are in each given frequency band within a certain frequency range. However, either the time-domain graph or the frequency-domain graph ignores one feature of the vibration signal. To address this issue, time-frequency analysis studies the signal in both the time and frequency domains simultaneously and produces time-frequency representations of the vibration signals. Therefore, two different representations were used to represent the pavement vibration signals, including the time-domain graph and the time-frequency graph. The former is the most commonly used method for representing vibration signals, while the latter can study the vibration signals in both the time and frequency domains simultaneously. Based on these two different representations, two different deep transfer learning-based methods would be proposed and compared for vehicle classification, which will be discussed in detail below.

The continuous wavelet transform is an excellent tool for time-frequency analysis, which has already attracted countless attention in the past decades. The continuous wavelet transform generates the time-frequency representation of the signal and plots it in the time-frequency graph (Fig. 4b). In the time-frequency graph, each data point corresponds to a certain time (X-axis) and frequency (Y-axis), while its color represents the absolute value of the wavelet coefficients. In this work, the continuous wavelet transform was performed by the complex Gaussian wavelet 8 (cgau8), while all these were based on a Python library named 'PyWavelets' [35].

2.3.2. Deep transfer learning

For the image classification task (e.g. vehicle classification in this work), deep transfer learning applied to CNN has two main steps (Fig. 5): (1) For the source domain, the CNN model is trained with a very larger dataset, usually, ImageNet [37] which has 1.2 million images with 1000 categories. The trained CNN model is often referred to as the pre-trained CNN model. (2) For the target domain, the pre-trained CNN model is used to accomplish the target task (vehicle classification in this work). In this process, the convolutional layers of the pre-trained CNN model (i.e., pre-trained convolutional layers) are retained, but the fully connected layers (i.e., FC layers) need to be changed to meet the requirements of the target task. As there are 1000 labels in the source domain and 10 labels in the target domain in this work, the output of the fully connected layers needs to be changed from 1000 to 10. After that, the adjusted pre-trained CNN model is fine-tuned with the dataset of the target domain (i.e., the dataset of asphalt pavement vibration signals).

EfficientNet [38] was used as the CNN model for deep transfer learning in this work. It balances network depth, width, and resolution to have a better performance than traditional CNN models [38]. With the relatively low computation complexity and model complexity, EfficientNet has a satisfactory performance in computer vision tasks. EfficientNet achieved a top-1 accuracy of 84.3% and a top-5 accuracy of 97.0% in ImageNet image classification [38]. Until now, the top-ranked CNN models in the ImageNet image classification task are still based on EfficientNet. Therefore, pre-trained EfficientNets (-B0, -B1, -B2, -B3, -B4, -B5, -B6, and -B7) were used for vehicle classification in this work, which are available on the online sources [39,40].

2.3.3. Evaluations metrics

Several evaluation metrics are commonly used to evaluate the performance of the CNN model for the image classification task. They include accuracy (Eq.(5)), precision (Eq.(6)), recall (Eq.(7)), and F_1 score (Eq.(8)). To consider the effect of multiple classes which are common in the image classification task, the three accuracy metrics have three subsets, 'macro', 'weighted', and 'micro'. 'Macro' means that it

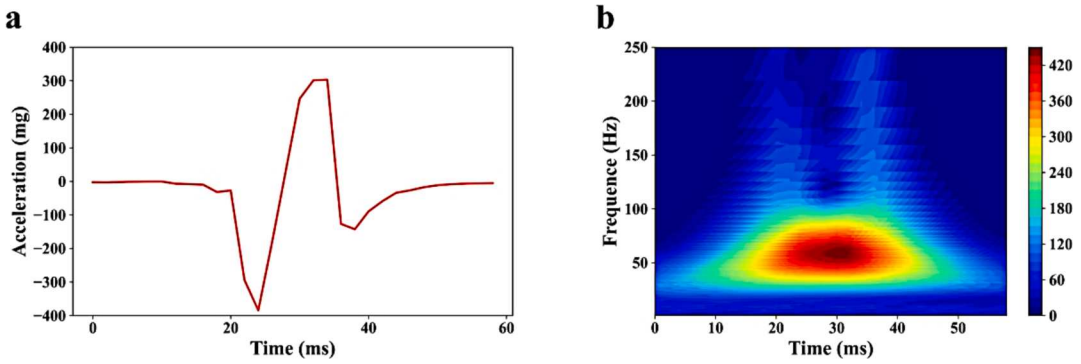

Fig. 4. Overview of (a) the time-domain graph and (b) the time-frequency graph.

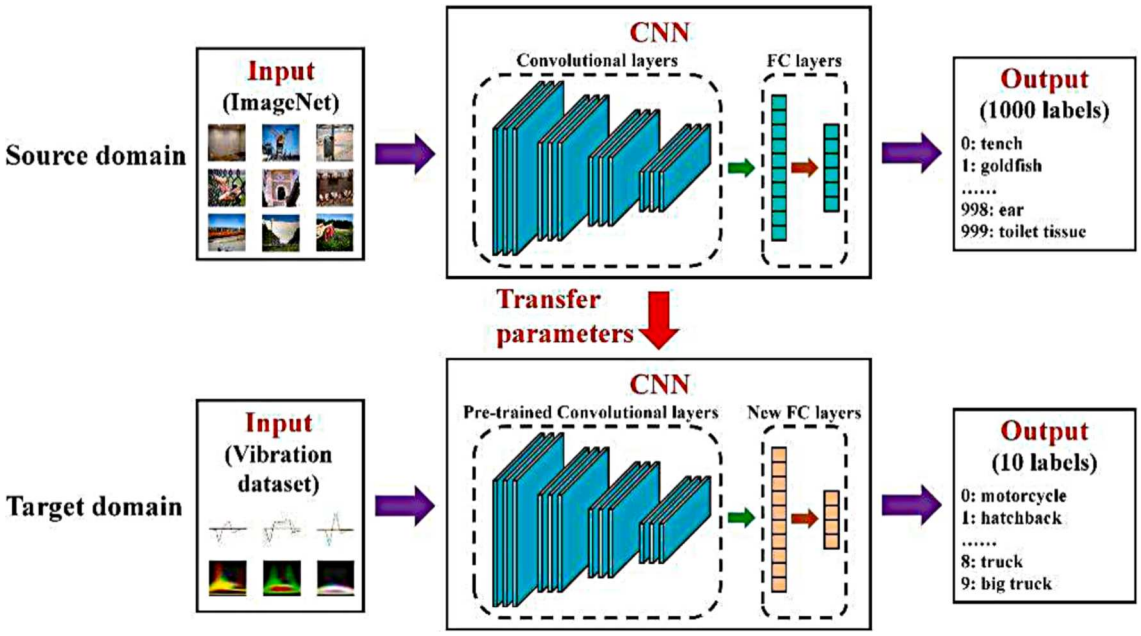

Fig. 5. Scheme of deep transfer learning applied to CNN.

needs to calculate accuracy metrics for each class and then obtain unweighted averages, regardless of label imbalances [41]. Similar to'macro', but 'weighted' takes into account label imbalances and calculates weighted averages [41]. In comparison, 'micro' calculates accuracy metrics by counting the total true positives, false negatives, and false positives, and thus the value of micro precision, micro recall, micro F_1 score, and accuracy are identical.

- Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- Precision

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

- Recall

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

- F_1 score

$$F_1 = \frac{2 \bullet Precision \bullet Recall}{Precision + Recall} \tag{8}$$

- Macro Precision

$$Precision_{macro} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l)}{|L|} \tag{9}$$

- Macro Recall

$$Recall_{macro} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l)}{|L|} \tag{10}$$

- Macro F_1 score

$$F_{1,\,macro} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l)}{|L|} \tag{11}$$

- Weighted Precision

$$Precision_{weighted} = \frac{\sum_{l \in L} Precision(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \tag{12}$$

- Weighted Recall

$$Recall_{weighted} = \frac{\sum_{l \in L} Recall(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \tag{13}$$

- Weighted F_1 score

$$F_{1,\,weighted} = \frac{\sum_{l \in L} F_1(y_l, \hat{y}_l) \times |y_l|}{\sum_{l \in L} |y_l|} \tag{14}$$

where L is the set of labels. y is the set of *true* (*sample, label*) pairs and y_l is the subset of y with label l. $\hat{y}$ is the set of *predicted* (*sample, label*) pairs and $\hat{y}_l$ is the subset of $\hat{y}$ with label l. *TP* is true positive, *TN* is true negative, *FP* is false positive, and *FN* is false negative.

2.3.4. Image preprocessing and hyperparameters

Image preprocessing is the step of formatting images before images are used for model training. The aim of image preprocessing is to improve the image and make it more suitable for deep learning models. In this work, resizing and normalization, two methods of image preprocessing, are applied to all images. In addition, as a form of image preprocessing, image data augmentation creates different versions of similar content so that the deep learning model is exposed to more training images to avoid the overfitting issue. Image data augmentation used in this work includes random cropping (RandomCrop), random horizontal flipping (RandomHorizontalFlip), and random rotation (RandomRotation). They are only applied in the training set.

The CNN model was based on the online sources [39,40] and run on PyTorch. The optimization algorithm is Adam [42] and the loss function is the cross-entropy loss. The input image size is 224 × 224 which meets the requirements of the pre-trained CNN model. Other hyperparameters include the learning rate (0.001), training epoch (100), and batch size

(32).

3. Results

3.1. Denoised vibration signals

As mentioned in Section 2.2, various threshold functions and wavelet functions were used in the wavelet transform for signal denoising. Table 1 summarizes the evaluation metrics of the wavelet transform for signal denoising. In three thresholding functions, the compromising method of hard and soft thresholding had the highest SNR and lowest RMSE. Its SNR was more than twice that of the hard thresholding and over three times that of the soft thresholding, while its RMSE is about one-third that of the hard and soft thresholding for Direction A. Although all wavelet functions had similar SNR and RMSE, 'bior6.8′ had the highest SNR and lowest RMSE. Therefore, the compromising method of hard and soft thresholding and 'bior6.8′ were finally selected as the threshold function and wavelet function.

To visualize the performance of signal denoising, Fig. 6 shows the images of the raw signals and denoised signals. As shown in Fig. 6a, the raw signals had numerous noises and after signal denoising, there were very few noises in the denoised signals which retained the meaningful signal. In the raw signal of acceleration sensing nodes, noises were mainly presented at low signal values (fluctuations at the beginning or end in Fig. 6b) and the signals with high absolute values (peaks or valleys in Fig. 6b) had few noises, which are more important for vehicle classification. In the denoised signals of acceleration sensing nodes, the most important signals (peaks or valleys in Fig. 6b) were almost retained in the original value while noises were almost removed (smoothed curves at the beginning or end in Fig. 6b). According to Table 1 and Fig. 6, the wavelet transform had a very satisfactory performance in signal denoising, which is important for signal preprocessing.

3.2. Datasets for deep transfer learning

After denoising signals, it needs to build the dataset for deep transfer learning to classify vehicles. As mentioned in Section 2.3, two methods were proposed for vehicle classification based on deep transfer learning. These two methods differed in the type of the input image, the time-domain graph and the time-frequency graph. Fig. 7a shows the time-domain graph of seven acceleration sensing nodes for one passing vehicle. The time-domain graph allows the vibration signals of all acceleration sensing nodes to be plotted in one graph, which could be found by the legend from 'Node-1' to 'Node-7'. In comparison, the time-frequency graph requires a single graph to plot the vibration signals of only one acceleration sensing node. It means that one passing vehicle needs seven time-frequency graphs to present the vibration signals of all acceleration sensing nodes. Fig. 7b shows three of the seven time-frequency graphs for the one passing vehicle in the same manner as Fig. 7a. In Fig. 7a, Node-6 had the highest vibration signals, followed by Node-3 and the other acceleration sensing nodes had very few vibration signals. Accordingly, Node-6 had the highest and most variable wavelet coefficients, and in Fig. 7b, it had the most colorful time-frequency graph. The wavelet coefficients of Node-3 were much lower than those of Node-6 and varied in a smaller ranger, so the time-frequency graph of Node-3 was less colorful with a few colors. In comparison, the time-frequency graph of Node-7 was plotted in blue as its vibration signals were almost zero.

Table 1
Evaluation metrics of the wavelet transform for signal denoising.

Threshold function	Wavelet function	Direction A		Direction B	
		SNR	RMSE	SNR	RMSE
Hard thresholding	bior6.8	4.0702	1.7223	3.1715	0.9486
Soft thresholding	bior6.8	2.6665	2.1132	2.0028	1.1278
Compromising method of hard and soft thresholding	bior6.8	**9.3397**	**0.6010**	**4.3913**	**0.5754**
	sym10	9.2976	0.6032	4.3494	0.5770
	sym8	9.3029	0.6029	4.3567	0.5769
	sym6	9.3104	0.6023	4.3540	0.5768
	coif5	9.2897	0.6035	4.3548	0.5768
	coif3	9.3093	0.6025	4.3567	0.5767
	rbio6.8	9.2801	0.6043	4.3421	0.5779
	db10	9.2304	0.6056	4.2475	0.5798
	db8	9.2401	0.6053	4.2641	0.5792
	db6	9.2736	0.6038	4.2914	0.5785
	db4	9.2849	0.6034	4.3116	0.5782

Note: 'bior' is Biorthogonal, 'sym' is Symlets, 'coif' is Coiflets, 'rbio' is Reverse biorthogonal, and 'db' is Daubechies.

When building the dataset, two methods need two different datasets. Method I (Time-domain & TL) is much easier to build its dataset because it only needs one time-domain graph to plot the vibration signals of all acceleration sensing nodes for one passing vehicle. However, Method II (Time-frequency & TL) needs seven time-frequency graphs to present the same corresponding information. How to combine these seven graphs to classify one passing vehicle? As shown in Fig. 7a, not all acceleration sensing nodes observed vibration signals. Therefore, the number of acceleration sensing nodes successfully obtaining vibration signals for one passing vehicle was summarized in Fig. 8. A maximum of three acceleration sensing nodes was employed to measure vibration signals for all vehicles. Only trucks and big trucks needed three acceleration sensing nodes and two acceleration sensing nodes were required for the other vehicles except for motorcycles which used one. It means three time-frequency graphs were needed instead of seven graphs to illustrate the vibration signals of all acceleration sensing nodes for one passing vehicle. However, the three time-frequency graphs were numbered differently for different vehicles because vehicles often passed through different locations of one lane. How to select these three time-frequency graphs? When selecting the three time-frequency graphs, it was necessary to choose the time-frequency graphs of acceleration sensing nodes that observed vibration signals. Therefore, it needed to first sort the acceleration sensing nodes by the absolute value of the observed vibration signals and then select the top three acceleration sensing nodes to plot their time-frequency graphs. These three graphs would be used for vehicle classification.

As there are three time-frequency graphs for one passing vehicle, it is possible to consider one graph as one channel of an RGB image. In particular, the time-frequency graph was changed from the RGB image (3 channels) to the gray image (1 channel) and then three time-frequency graphs were combined together into an RGB image. For example, three time-frequency graphs in Fig. 7b were combined to plot an RGB image in Fig. 9b. This image would be used to build the dataset for Method II (Time-frequency & TL). In order to eliminate irrelevant information, the axis ticks, axis labels, legends, titles, color bars, etc. were removed during the building of the datasets. Fig. 9 shows the exemplary images of the datasets for the two methods.

Table 2 summarizes the information of the dataset built in this work. This dataset contained 10 categories of vehicles, including the motorcycle, hatchback, sedan, SUV, mini VAN, VAN, pickup truck, mini truck, truck, and big truck. The number of each category varied a lot and there was a total of 2098 vehicles collected in this dataset. The number of sedans was the largest, followed by the truck and big truck, which was much larger than the other vehicle types. To train and evaluate the CNN model, the whole dataset was separated into two subsets, training (80%) and test (20%) sets. The number of vehicles in each category in the training and test sets could also be found in Table 2.

3.3. Accuracy metrics of vehicle classification

3.3.1. Accuracy metrics of image preprocessing methods

As mentioned above, image data augmentation exposes the CNN model to more complex training images to avoid the overfitting issue. Three image data augmentation methods, including RandomCrop, RandomHorizontalFlip, and RandomRotation, were used in this work.

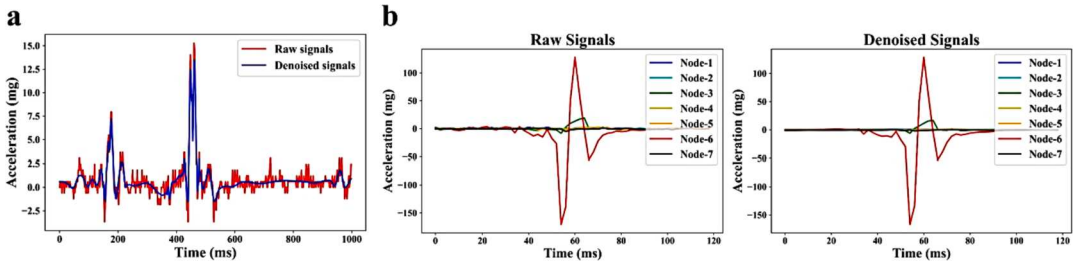

Fig. 6. Comparison of raw signals and denoised signals. (a) Raw signals and denoised signals of one acceleration sensing node and (b) raw signals and denoised signals of seven acceleration sensing nodes.

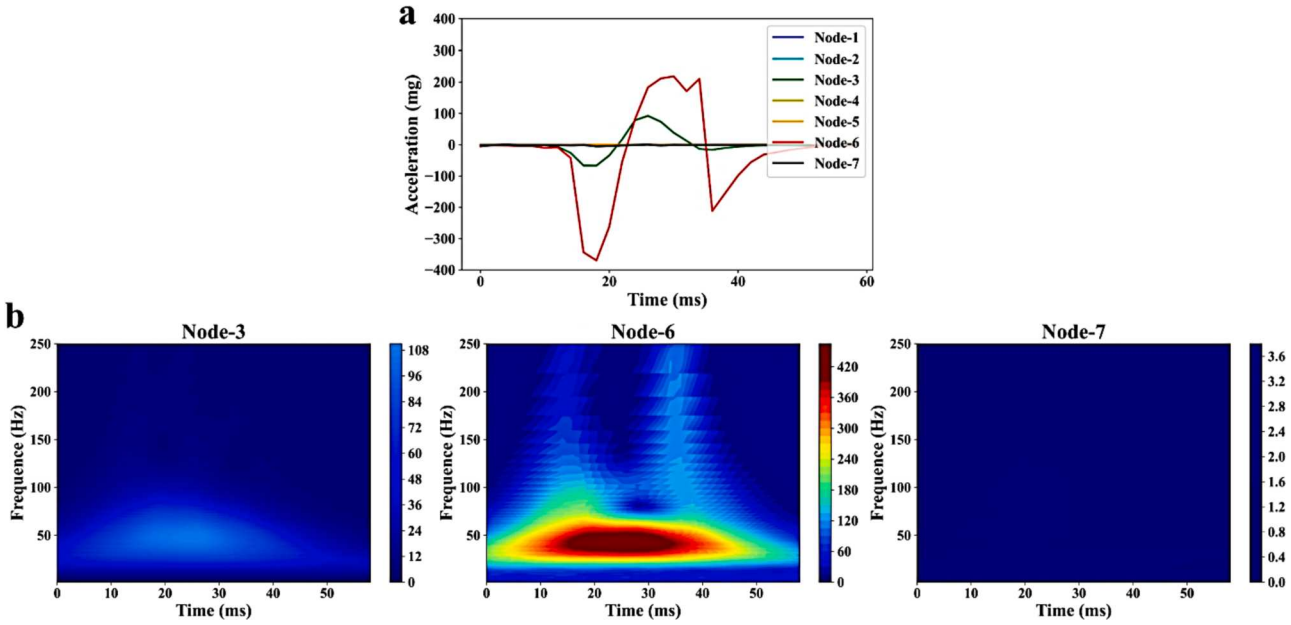

Fig. 7. Denoised signals of seven acceleration sensing nodes and their corresponding time-frequency graphs. (a) Denoised signals of seven acceleration sensing nodes and (b) time-frequency graphs of Node-3, Node-6, and Node-7.

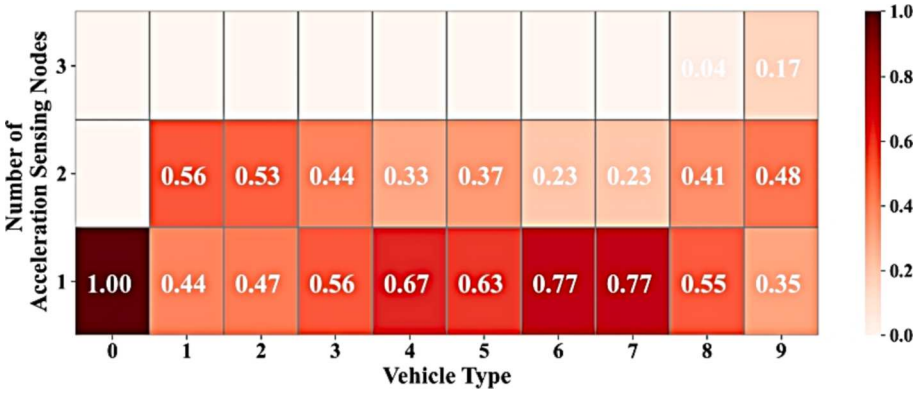

Fig. 8. The number of acceleration sensing nodes that received effective vibration signals (defined in [8,13]). 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

Table 3 summarizes the accuracy metrics of the test set in terms of image preprocessing methods. Without image data augmentation, the CNN model in both methods faced the overfitting issue. After applying image data augmentation, the accuracy of the CNN model improved a lot no matter which method was chosen. So it is necessary to apply image data augmentation to the training set. In addition, the same image data augmentation method differently affected the accuracy of the two methods. RandomCrop had the highest accuracy in Method I and RandomHorizontalFlip plus RandomRotation achieved the highest accuracy in Method II. Therefore, Method I and Method II employed different image data augmentation methods to maximize their performance. After applying the selected image data augmentation methods to the training set, the pre-trained CNN models for Method I and Method II were trained and evaluated on the dataset for vehicle classification.

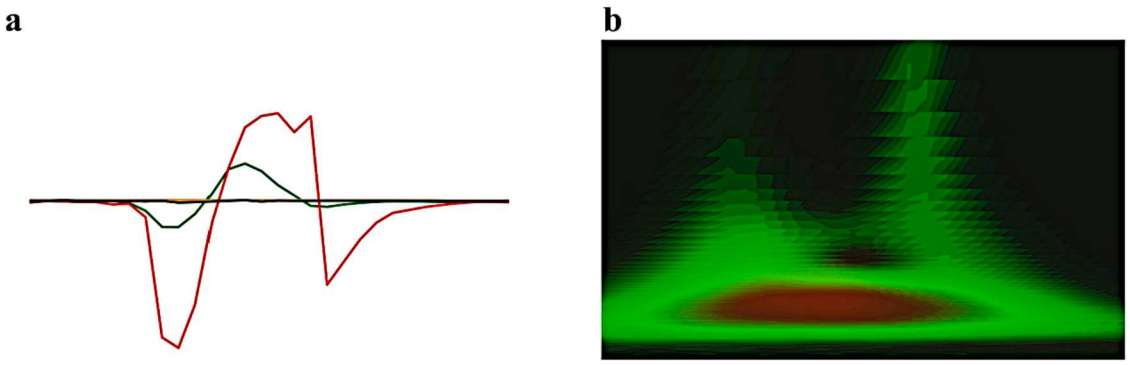

Fig. 9. Image of the dataset for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL).

Table 2
Summary of the dataset.

	0	1	2	3	4	5	6	7	8	9	Total
Train	32	40	443	206	169	50	18	38	390	301	1687
Test	4	14	133	49	30	7	4	6	56	108	411
Total	36	54	576	255	199	57	22	44	446	409	2098

Note: 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

Table 3
Accuracy metrics of image preprocessing methods.

Image preprocessing	Method I (Time-domain & TL)			Method II (Time-frequency & TL)		
	Accuracy	Macro F_1	Weighted F_1	Accuracy	Macro F_1	Weighted F_1
Resize	0.8200	0.6431	0.8103	0.8856	0.7731	0.8818
Resize RandomCrop	**0.9489**	**0.9299**	**0.9475**	0.9416	0.8089	0.9372
Resize RandomHorizontalFlip RandomRotation	0.9270	0.8836	0.9259	**0.9611**	**0.8738**	**0.9578**
Resize RandomCrop RandomHorizontalFlip	0.9319	0.8516	0.9286	0.9246	0.7176	0.9148
Resize RandomCrop RandomHorizontalFlip RandomRotation	0.9173	0.8392	0.9146	0.9221	0.7390	0.9155

3.3.2. *Accuracy metrics of deep transfer learning*

Two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for vehicle classification. Fig. 10 shows the loss of these two methods. Three EfficientNets (-B0, -B3, and -B7) were selected to present their results because these three CNN models have different accuracy on ImageNet with different computational and model complexity. For Method I (Fig. 10a), the loss of three CNN models had a similar pattern that the loss decreased with the increasing epoch and then converged despite few fluctuations. The models in Method II performed similarly to those in Method I. However, these three CNN models in Method II converged at approximately Epoch-40, much faster than those in Method I (about Epoch-65), while the final loss values of

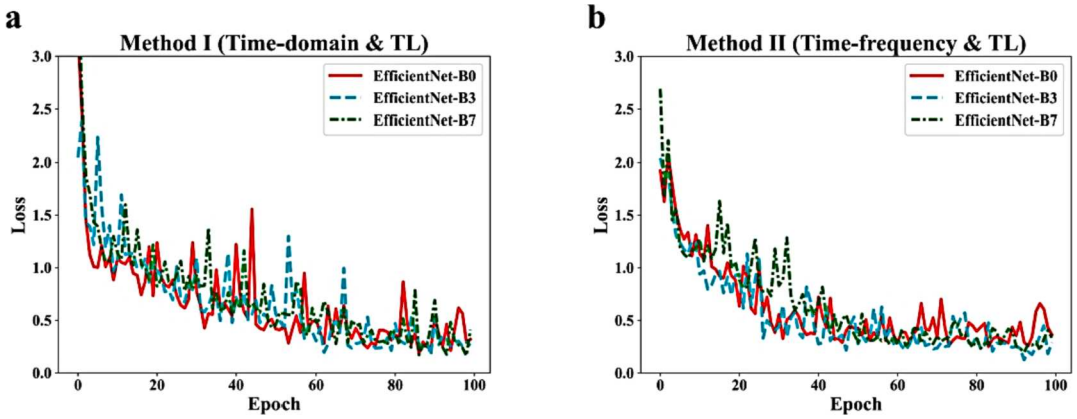

Fig. 10. Loss of two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL).

the three CNN models were similar in these two methods.

Fig. 11 shows the accuracy metrics of two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). In summary, similar to the loss (Fig. 10), the CNN models in Method II converged much faster than those in Method I. Although three CNN models eventually reached similar values for all accuracy metrics in both methods, the accuracy metrics of EfficientNet-B0 and -B3 increased more rapidly than EfficientNet-B7, especially in Method II. In both two methods, all three CNN models had similar patterns between the accuracy and weighted F_1 score, which was higher than the macro F_1 score.

Table 4 summarizes the accuracy metrics of Method I (Time-domain & TL). EfficientNet-B3 achieved the highest values for all accuracy metrics, except for the macro precision and macro F_1 score, which were the highest in EfficientNet-B0. It means that considering the label imbalance, EfficientNet-B3 had the best performance, but without eliminating the effect of the label imbalance, EfficientNet-B0 performed better than EfficientNet-B3. Table 5 summarizes the accuracy metrics of Method II (Time-frequency & TL). Similar to the result of Method I, EfficientNet-B3 had the highest values for all accuracy metrics, except for the macro precision, macro recall, and macro F_1 score. EfficientNet-B2 achieved the highest macro precision and EfficientNet-B7 did it in the macro recall and macro F_1 score. In both two methods, EfficientNet-B3 had the highest accuracy metrics when considering the label imbalance, but other EfficientNets performed better in the accuracy metrics without taking into account the label imbalance. Meanwhile, EfficientNet had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95, indicating that the CNN model with deep transfer learning could be an effective, accurate, and reliable tool for vehicle classification based on asphalt pavement vibration. From the perspective of accuracy metrics, EfficientNet-B3 is the best model in both methods.

In addition, there is no obvious relationship between the accuracy and the computation and model complexity. For example, EfficientNet-B2 and EfficientNet-B7 had identical accuracy in Method I while the computation and model complexity of EfficientNet-B7 were much more complex than EfficientNet-B2. In Method II, the accuracy of EfficientNet-B2 was much higher than that of EfficientNet-B7.

4. Discussion

Two deep transfer learning-based methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL), were proposed for vehicle classification in this work and the results of both methods show that the CNN model with deep transfer learning could be an effective, accurate,

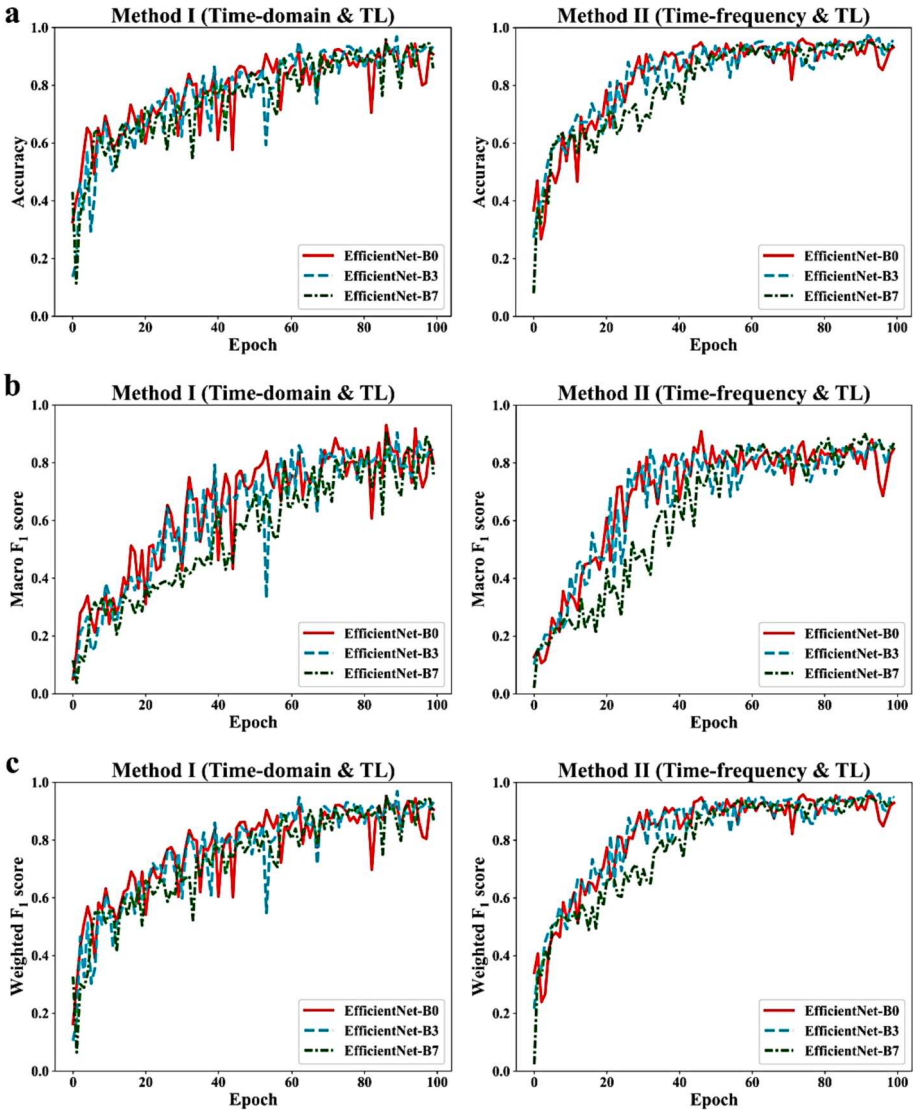

Fig. 11. Accuracy metrics of two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

Table 4
Accuracy metrics of Method I (Time-domain & TL).

Model	Accuracy	Precision		Recall		F_1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
EfficientNet-B0	0.9489	**0.9714**	0.9503	0.9022	0.9489	**0.9299**	0.9475
EfficientNet-B1	0.9513	0.9410	0.9545	0.8982	0.9513	0.9082	0.9509
EfficientNet-B2	0.9586	0.9533	0.9608	0.8840	0.9586	0.9046	0.9571
EfficientNet-B3	**0.9684**	0.9247	**0.9728**	**0.9099**	**0.9684**	0.9042	**0.9688**
EfficientNet-B4	0.9635	0.9518	0.9653	0.8868	0.9635	0.8941	0.9615
EfficientNet-B5	0.9635	0.9437	0.9637	0.8359	0.9635	0.8761	0.9618
EfficientNet-B6	0.9562	0.9534	0.9601	0.8819	0.9562	0.8934	0.9546
EfficientNet-B7	0.9586	0.9631	0.9606	0.8819	0.9586	0.9097	0.9569

Table 5
Accuracy metrics of Method II (Time-frequency & TL).

Model	Accuracy	Precision		Recall		F_1 score	
		Macro	Weighted	Macro	Weighted	Macro	Weighted
EfficientNet-B0	0.9611	0.9765	0.9626	0.8356	0.9611	0.8738	0.9578
EfficientNet-B1	0.9611	0.9740	0.9619	0.8501	0.9611	0.8918	0.9590
EfficientNet-B2	0.9708	**0.9844**	0.9715	0.8529	0.9708	0.8909	0.9680
EfficientNet-B3	**0.9732**	0.9702	**0.9744**	0.8340	**0.9732**	0.8673	**0.9700**
EfficientNet-B4	0.9684	0.9735	0.9692	0.8212	0.9684	0.8514	0.9641
EfficientNet-B5	0.9635	0.9752	0.9646	0.8293	0.9635	0.8679	0.9604
EfficientNet-B6	0.9659	0.9628	0.9664	0.8691	0.9659	0.8916	0.9637
EfficientNet-B7	0.9562	0.9698	0.9571	**0.8841**	0.9562	**0.9035**	0.9541

and reliable technique for vehicle classification based on asphalt pavement vibration. These two methods employed different input types and performed differently on the accuracy metrics. So what is the difference between these two methods? What are the advantages and disadvantages of these two methods? These two methods differ in three aspects:

4.1. Difference in input types

These two methods utilized different inputs, the time-domain graph and the time-frequency graph. Their inputs are different in three ways. (1) The features of vibration signals. Method I only used time features of vibration signals and Method II processed both time and frequency features of vibration signals simultaneously. (2) The dimension of inputs. Although both methods employed graphs, the dimension of inputs is different. The time-domain graph is two-dimensional, while the X-axis is time and the Y-axis is the vibration signal (Fig. 4a). The time-frequency graph could be viewed as three-dimensional, while the X-axis is time, the Y-axis is frequency, and the Z-axis (color) is the wavelet coefficients (Fig. 4b). (3) The data fusion method. As seven acceleration sensing nodes were used to measure pavement vibration for one passing vehicle, it needs to combine the vibration signals of all acceleration sensing nodes as the input to the CNN model. In Method I, the vibration signals of all acceleration sensing nodes are easily plotted in an image that is considered as the input (Fig. 9a). In comparison, in Method II, each acceleration sensing node has its own time-frequency graph. After statistical analysis (Fig. 8), the top three acceleration sensing nodes are selected to plot their time-frequency graphs as gray (1 channel) images. Then these three gray images are combined as one RGB (3 channels) image that is considered as the input (Fig. 9b).

4.2. Difference in accuracy metrics

To better compare the two methods, Fig. 12 summarizes the accuracy metrics of the CNN models for two methods and Fig. 13 presents the normalized confusion matrix of EfficientNet-B0 and -B3 for two methods. In terms of accuracy metrics, these two methods differ in three ways.

(1) The accuracy considers the label imbalance. The label imbalance means that the distribution of images across classes is not identical. In the label imbalance, there are two groups, majority class and minority class, while the former has many images and the latter has few images. The mass of images from the majority class would drown out the minority class, resulting in the deep learning model focusing only on learning the feature of the majority class with abundant images and ignoring the minority class which is in fact equally meaningful [43,44]. This is because the deep learning model assumes that each class has an equal distribution [44]. The dataset in this work had ten classes with different numbers of images (Table 2), resulting in the label imbalance. The accuracy, weighted precision, weighted recall, and weighted F_1 score take into account the label imbalance. As shown in Fig. 12a and c, the accuracy and weighted F_1 score of almost all CNN models in Method II were higher than those in Method I. When back to the confusion matrix shown in Fig. 13, compared to Method I, the CNN models in Method II have better performance in the majority class. This is because the important informative region in the time-frequency graph is larger than that of the time-domain graph (Fig. 14a). Also, the discrepancy between similar vibration signals is more apparent in the time-frequency graph compared to the time-domain graph (Fig. 14b and c). These characteristics help make the CNN model easier to learn from the time-frequency graph, thus improving accuracy.

(2) The accuracy without taking into account the label imbalance. In comparison, the macro precision, macro recall, and macro F_1 score do not account for the label imbalance and assume that each label is equally important. As shown in Fig. 12b, the macro F_1 score of all CNN models in Method I was higher than those in Method II. This is due to the fact that the CNN models in Method I perform better in the minority class (e.g. motorcycle and pickup truck) than Method II (Fig. 13) and the unweighted mean between the majority class and minority class improves the final value of accuracy metrics. The minority class has few examples, making the CNN models in Method II more difficult to be trained because the time-frequency graph is more complex than the time-domain graph. In addition, different time-domain graphs might have similar time-frequency graphs (Fig. 14d), especially for the motorcycle that collects vibration signals with one acceleration sensing node. This led to the low accuracy of the CNN models in Method II for the motorcycle (Fig. 13).

(3) The convergence speed and stability of the CNN model. As shown in Fig. 11, the CNN models in Method II converged faster and performed more stably than those in Method I by using the same hyperparameters. This is important for training the CNN model. The rapid convergence

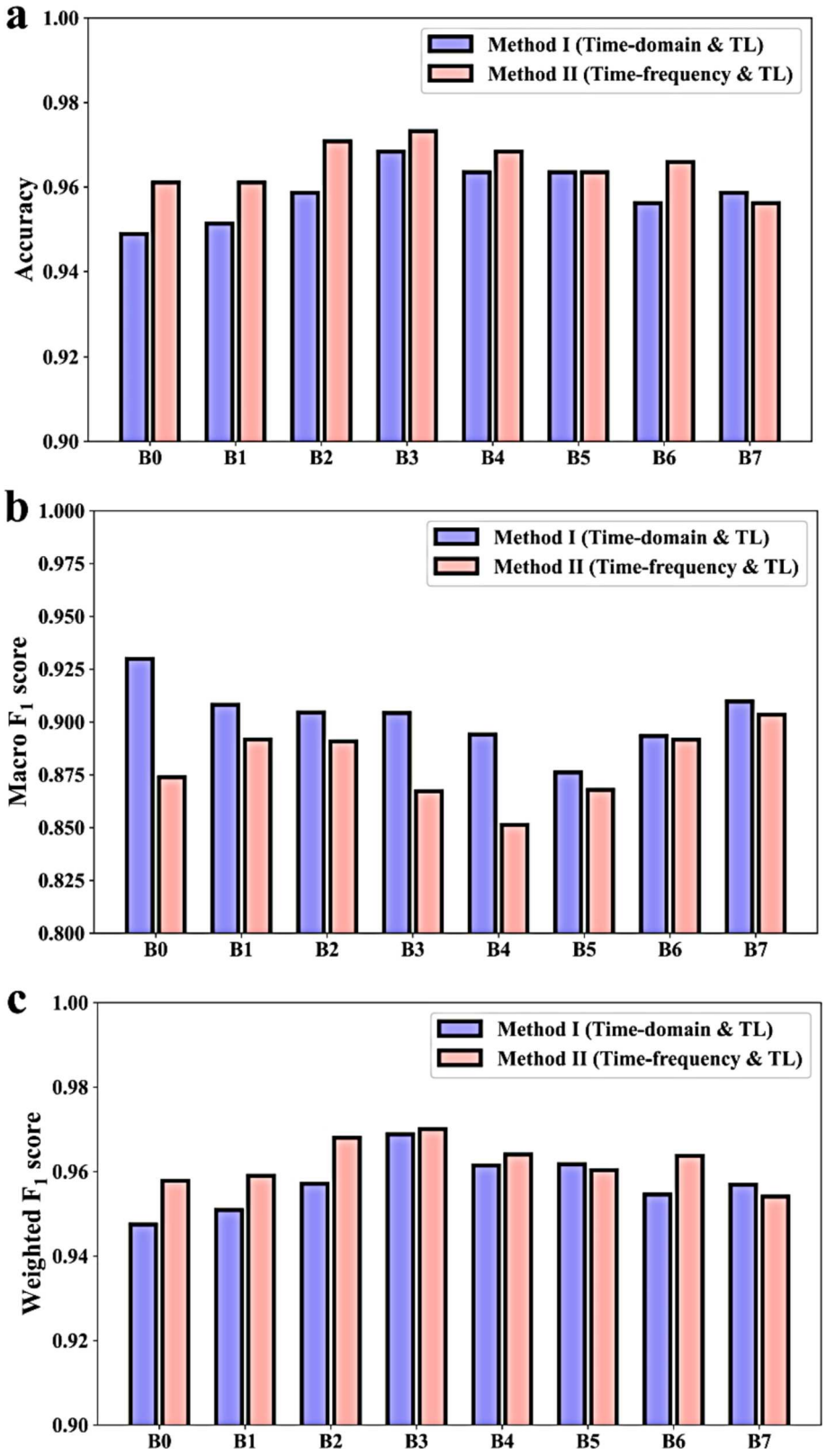

Fig. 12. Accuracy metrics of the CNN models for two methods, Method I (Time-domain & TL) and Method II (Time-frequency & TL). (a) Accuracy, (b) macro F_1 score, and (c) weighted F_1 score.

means that the CNN model needs fewer epochs or training time and the high stability means that the CNN model is more robust.

4.3. Difference in application prospects

These two methods also have different application prospects. (1) Method I runs at a higher running speed than Method II and requires less

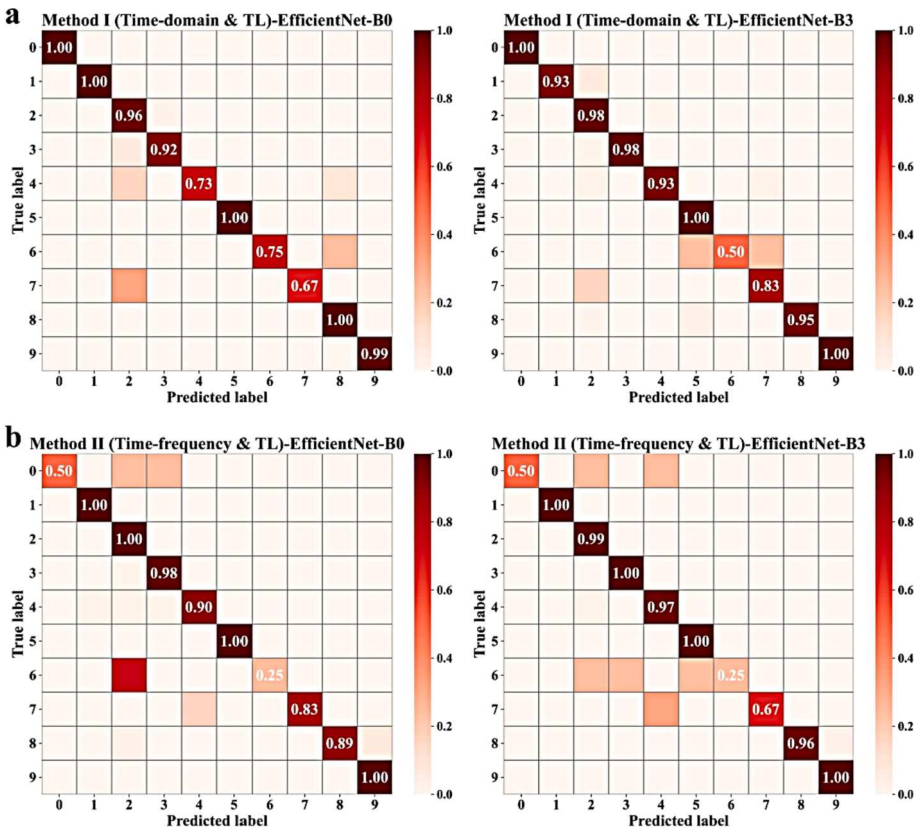

Fig. 13. Normalized confusion matrix of EfficientNet-B0 and -B3 for two methods. (a) Method I (Time-domain & TL) and (b) Method II (Time-frequency & TL). Note: 0: motorcycle, 1: hatchback, 2: sedan, 3: SUV, 4: mini VAN, 5: VAN, 6: pickup truck, 7: mini truck, 8: truck, and 9: big truck.

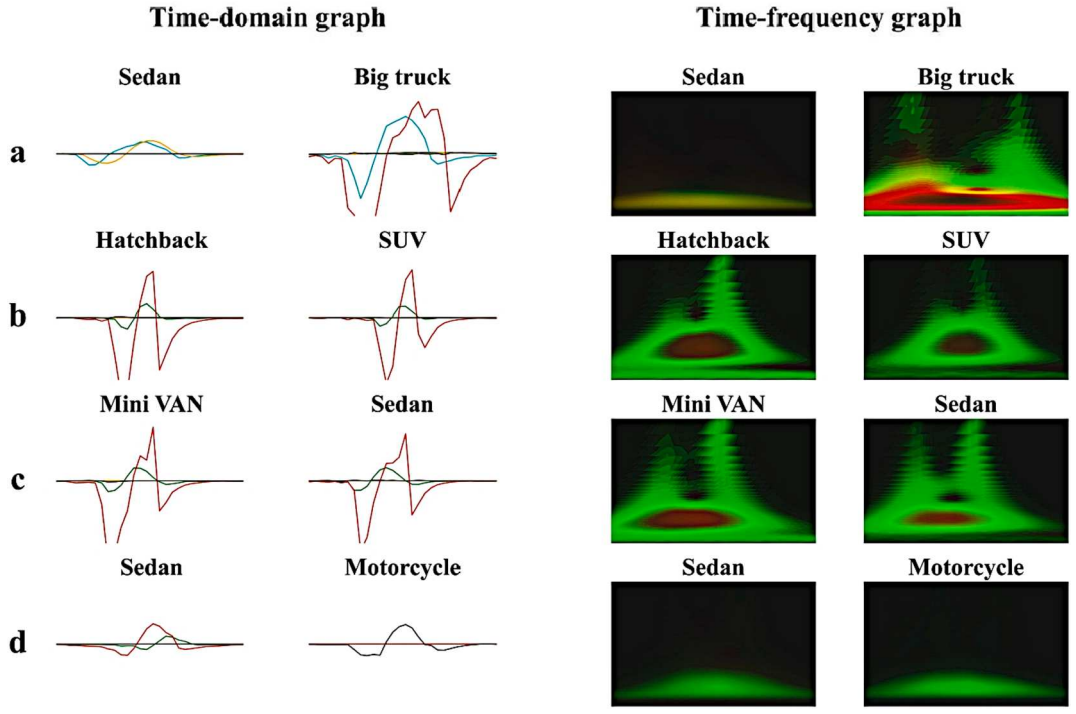

Fig. 14. Comparison of time-domain graphs and time-frequency graphs.

time for vehicle classification. Compared to Method I, Method II introduces the time-frequency analysis to obtain the time-frequency graph and although the time-frequency analysis is very fast, it still takes some time. The high usage time for vehicle classification might limit the application of Method II in contrast to Method I. (2) Method II needs more data (image) than Method I. According to Fig. 12 and Fig. 13,

compared with Method I, the CNN model in Method II performed better in the majority class, but worse in the minority class. The time-frequency graph is more complex than the time-domain graph (Fig. 14), so the CNN model in Method II requires more data to maximize its performance. In the future, more data will be collected on each vehicle type to increase the dataset size and eliminate the label imbalance in the dataset. (3) Method II employs a different data fusion method than Method I. Unlike Method I, which plots the vibration signals of all acceleration sensing nodes in one graph, Method II requires several graphs to present the same information about one passing vehicle as in Method I. In this work, after statistics analysis, three time-frequency graphs were combined as the input. But if the acceleration sensing nodes are arranged too close together or the vehicle is too wide, more than three acceleration sensing nodes might collect the vibration signal. How to fuse data? There are two alternatives. The first method is to increase the channel of the input to combine all possible time-frequency graphs. If the input has more than three channels, the first convolutional layer of the pre-trained CNN model needs to be adjusted, which may affect its performance. Another method is to use the results of the time-frequency analysis as the matrix rather than the graph and then apply the LSTM to process the matrix. This method was also investigated by previous literature [45,46]. The problem with this method is that it is difficult to find the pre-trained LSTM with large datasets, which is different from the pre-trained CNN model. In this case, the LSTM might not achieve a similar performance to the pre-trained CNN model due to the lack of data.

5. Conclusion

Deep transfer learning needs a small number of labeled data to achieve satisfactory performance in the target domain, and thus has great potential for a wide range of applications in civil engineering. This work aims to propose a deep transfer learning-based method for vehicle classification by asphalt pavement vibration. The main contributions of this work include: (1) the wavelet transform was used to denoise the pavement vibration signals which were collected by the pavement vibration IoT monitoring system; (2) the pavement vibration signals were represented in two different ways, including the time-domain graph and the time-frequency graph; (3) two deep transfer learning-based methods were proposed for vehicle classification based on the two different representations of the pavement vibration signals. In addition, the input types, accuracy metrics, and application prospects were applied to investigate the feasibility and validity of two deep transfer learning-based methods for vehicle classification. The results show that the CNN model had a satisfactory performance in both methods with the accuracy of Method I exceeding 0.94 and Method II exceeding 0.95. The CNN model in Method II performed better in the accuracy metrics with considering label imbalance, but worse in the accuracy metrics without considering label imbalance than that in Method I. In both two methods, EfficientNet-B3 had the highest accuracy metrics when considering the label imbalance, but other EfficientNets performed better in the accuracy metrics without taking into account the label imbalance. The CNN model with deep transfer learning could be an effective, accurate, and reliable method for vehicle classification based on asphalt pavement vibration.

CRediT authorship contribution statement

Fangyu Liu: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Zhoujing Ye:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Linbing Wang:** Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

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.

Acknowledgements

This work was supported by a grant from the National Science Foundation (No. 1939987). This work was also sponsored by a grant from the Center for Integrated Asset Management for Multimodal Transportation Infrastructure Systems (CIAMTIS), a US Department of Transportation University Transportation Center, under federal grant number 69A3551847103. The authors are grateful for the support.

References

[1] N.K. Kanhere, S.T. Birchfield, A taxonomy and analysis of camera calibration methods for traffic monitoring applications, IEEE Transactions on Intelligent Transportation Systems 11 (2) (2010) 441–452.
[2] J. Sánchez-Oro, D. Fernández-López, R. Cabido, A.S. Montemayor, J.J. Pantrigo, Radar-based road-traffic monitoring in urban environments, Digital Signal Processing 23 (1) (2013) 364–374.
[3] J. Zhang, Y. Lu, Z. Lu, C. Liu, G. Sun, Z. Li, A new smart traffic monitoring method using embedded cement-based piezoelectric sensors, Smart Materials and Structures 24(2) (2015) 025023.
[4] B. Coifman, S. Kim, Speed estimation and length based vehicle classification from freeway single-loop detectors, Transportation research part C: emerging technologies 17 (4) (2009) 349–364.
[5] H. Zhao, D. Wu, M. Zeng, S. Zhong, A vibration-based vehicle classification system using distributed optical sensing technology, Transportation Research Record 2672 (43) (2018) 12–23.
[6] D. Kleyko, R. Hostettler, N. Lyamin, W. Birk, U. Wiklund, E. Osipov, Vehicle classification using road side sensors and feature-free data smashing approach, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 1988–1993.
[7] R. Bajwa, R. Rajagopal, P. Varaiya, R. Kavaler, In-pavement wireless sensor network for vehicle classification, in: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2011, pp. 85–96.
[8] Z. Ye, G. Yan, Y. Wei, B. Zhou, N. Li, S. Shen, L. Wang, Real-Time and Efficient Traffic Information Acquisition via Pavement Vibration IoT Monitoring System, Sensors 21 (8) (2021) 2679.
[9] Z. Ye, H. Xiong, L. Wang, Collecting comprehensive traffic information using pavement vibration monitoring data, Computer-Aided Civil and Infrastructure Engineering 35 (2) (2020) 134–149.
[10] W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller, P. Varaiya, A wireless accelerometer-based automatic vehicle classification prototype system, IEEE Transactions on Intelligent Transportation Systems 15 (1) (2013) 104–111.
[11] Y. Huang, L. Wang, Y. Hou, W. Zhang, Y. Zhang, A prototype IOT based wireless sensor network for traffic information monitoring, International journal of pavement research and technology 11 (2) (2018) 146–152.
[12] M.F. Duarte, Y.H. Hu, Vehicle classification in distributed sensor networks, Journal of Parallel and Distributed Computing 64 (7) (2004) 826–838.
[13] Z. Ye, L. Wang, W. Xu, Z. Gao, G. Yan, Monitoring traffic information with a developed acceleration sensing node, Sensors 17 (12) (2017) 2817.
[14] Z. Ye, Y. Lu, L. Wang, Investigating the pavement vibration response for roadway service condition evaluation, Advances in Civil Engineering 2018 (2018) 1–14.
[15] Y. Yu, C. Wang, X. Gu, J. Li, A novel deep learning-based method for damage identification of smart building structures, Structural Health Monitoring 18 (1) (2019) 143–163.
[16] H.V. Dang, H. Tran-Ngoc, T.V. Nguyen, T. Bui-Tien, G. De Roeck, H.X. Nguyen, Data-driven structural health monitoring using feature fusion and hybrid deep learning, IEEE Transactions on Automation Science and Engineering 18 (4) (2020) 2087–2103.
[17] J. Pal, S. Sikdar, S. Banerjee, A deep-learning approach for health monitoring of a steel frame structure with bolted connections, Structural Control and Health Monitoring 29 (2) (2022) e2873.
[18] O. Alazzawi, D. Wang, A novel structural damage identification method based on the acceleration responses under ambient vibration and an optimized deep residual algorithm, Structural Health Monitoring 14759217211065009 (2022).
[19] F. Liu, W. Ding, Y. Qiao, L. Wang, An artificial neural network model on tensile behavior of hybrid steel-PVA fiber reinforced concrete containing fly ash and slag power, Frontiers of Structural and Civil Engineering 14 (6) (2020) 1299–1315.
[20] F.-Y. Liu, W.-Q. Ding, Y.-F. Qiao, L.-B. Wang, Q.-Y. Chen, Compressive behavior of hybrid steel-polyvinyl alcohol fiber-reinforced concrete containing fly ash and slag powder: experiments and an artificial neural network model, Journal of Zhejiang University-SCIENCE A 22 (9) (2021) 721–735.
[21] F. Liu, L. Wang, UNet-based model for crack detection integrating visual explanations, Construction and Building Materials 322 (2022), 126265.

[22] F. Liu, J. Liu, L. Wang, Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography, IEEE Transactions on Intelligent Transportation Systems (2022).
[23] Fangyu Liu, Jian Liu, Linbing Wang, Deep learning and infrared thermography for asphalt pavement crack severity classification, Automation in Construction 140 (2022), 104383.
[24] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.
[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, Springer, International conference on artificial neural networks, 2018, pp. 270–279.
[26] Y. Gao, K.M. Mosalam, Deep transfer learning for image-based structural damage recognition, Computer-Aided Civil and Infrastructure Engineering 33 (9) (2018) 748–768.
[27] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Construction and building materials 157 (2017) 322–330.
[28] H. Kim, H. Kim, Y.W. Hong, H. Byun, Detecting construction equipment using a region-based fully convolutional network and transfer learning, Journal of computing in Civil Engineering 32 (2) (2018) 04017082.
[29] Y. Li, T. Bao, H. Chen, K. Zhang, X. Shu, Z. Chen, Y. Hu, A large-scale sensor missing data imputation framework for dams using deep learning and transfer learning strategy, Measurement 178 (2021), 109377.
[30] Q. Pan, L. Zhang, G. Dai, H. Zhang, Two denoising methods by wavelet transform, IEEE transactions on signal processing 47 (12) (1999) 3401–3406.
[31] S.W. Katicha, A. Loulizi, J.E. Khoury, G.W. Flintsch, Adaptive false discovery rate for wavelet denoising of pavement continuous deflection measurements, Journal of computing in civil engineering 31 (2) (2017) 04016049.
[32] J. Baili, S. Lahouar, M. Hergli, I.L. Al-Qadi, K. Besbes, GPR signal de-noising by discrete wavelet transform, Ndt & E International 42 (8) (2009) 696–703.
[33] S. Guoxiang, Z. Ruizhen, Three novel models of threshold estimator for wavelet coefficients, International Conference on Wavelet Analysis and Its Applications, Springer (2001) 145–150.
[34] I.M. Johnstone, B.W. Silverman, Wavelet threshold estimators for data with correlated noise, Journal of the royal statistical society: series B (statistical methodology) 59 (2) (1997) 319–351.
[35] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, A. O'Leary, PyWavelets: A Python package for wavelet analysis, Journal of Open Source Software 4 (36) (2019) 1237.
[36] X. An, J. Yang, Denoising of hydropower unit vibration signal based on variational mode decomposition and approximate entropy, Transactions of the Institute of Measurement and Control 38 (3) (2016) 282–292.
[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248-255.
[38] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, International Conference on Machine Learning, PMLR, 2019, pp. 6105-6114.
[39] L. Melas-Kyriazi, EfficientNet PyTorch, 2021. https://github.com/lukemelas/EfficientNet-PyTorch.
[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026–8037.
[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.
[42] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[43] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera (Eds.), Learning from Imbalanced Data Sets, Springer International Publishing, Cham, 2018.
[44] J. Brownlee, Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning, Machine Learning Mastery (2020).
[45] M. Liu, S. Liao, Y. Yang, Y. Men, J. He, Y. Huang, Tunnel boring machine vibration-based deep learning for the ground identification of working faces, Journal of Rock Mechanics and Geotechnical Engineering 13 (6) (2021) 1340–1357.
[46] M. Rautela, S. Gopalakrishnan, Ultrasonic guided wave based structural damage detection and localization using model assisted convolutional and recurrent neural networks, Expert Systems with Applications 167 (2021), 114189.