A

Research Paper

On

Real time traffic monitoring system using deep learning

Submitted in partial fulfilment of the requirements

For the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Name: Km Shivani(2000970100058)

Name: Abhigyan(2000970100002)

Name: Aditya Singh(2000970100007)

Semester – VII

Under the Supervision of

Dr. Ramveer Singh

Galgotias College of Engineering & Technology

Greater Noida, Uttar Pradesh

India-20130

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University

Abstract

Currently, vehicle classification in roadway-based techniques depends mainly on pho tos/videos collected by an over-roadway camera or on the magnetic characteristics of vehicles. However, camera-based techniques are criticized for potentially violating the privacy of vehicle occu pants and exposing their identity, and vehicles can evade detection when they are obscured by larger vehicles. Here, we evaluate methods of identifying and classifying vehicles on the basis of seismic data. Vehicle identification from seismic signals is considered a difficult task because of interference by various noise. By analogy with techniques used in speech recognition, we used different artificial intelligence techniques to extract features of three, different-sized vehicles (buses, cars, motorcycles) and seismic noise. We investigated the application of a deep neural network (DNN), a convolutional neural network (CNN), and a recurrent neural network (RNN) to classify vehicles on the basis of vertical-component seismic data recorded by geophones. The neural networks were trained on 5580 unprocessed seismic records and achieved excellent training accuracy (99%). They were also tested on large datasets representing periods as long as 1 month to check their stability. We found that CNN was the most satisfactory approach, reaching 96% accuracy and detecting multiple vehicle classes at the same time at a low computational cost. Our findings show that seismic methods can be used for traffic monitoring and security purposes without violating the privacy of vehicle occupants, offering greater efficiency and lower costs than current methods. A similar approach may be useful for other types of transportation, such as vessels and airplanes.

Keywords: traffic monitoring; deep learning; convolutional neural network; recurrent neural net work; deep neural network; signal-to-noise ratio.

1.Introduction

Many countries invest heavily in traffic monitoring systems [1], which collect and analyze traffic data to derive statistical information, such as the numbers of vehicles on the road and their temporal patterns. Governments use these statistics to forecast transportation needs, improve transportation safety, and schedule pavement maintenance work. Identifying the size of vehicles is a key task that helps to predict noise levels and road damage. The characteristic mix of vehicle types that use a roadway can determine the geometric design of the road based on the Traffic Monitoring Guide report published by the Federal Highway Administration in the United States [2]. Vehicle classification systems make use of many recent advances in sensing and ma chine learning technologies [3]. Although newer systems perform vehicle classification with higher accuracy, they differ in their characteristics and requirements, such as the types of sensors used, parameter settings, operating environment, and cost. Many traffic monitoring systems rely on vision-based vehicle classification techniques, usually based on cameras, that deliver high classification accuracy ranged between 90%~99% [4], covering large areas compared with emerging alternatives. Although camera-based systems have high classification accuracy, their performance can be affected by weather and lighting conditions as well as other factors. For instance, vehicles can be missed when they are obscured by large vehicles. Furthermore, the system requires huge investments in infrastructures to perform a complete coverage of the road network. Another important problem is the privacy concerns of vehicle occupants, as many people do not feel comfortable being exposed to cameras. An inductive loop detector based on magnetic characteristics of vehicles is one of the most commonly used traffic monitoring systems for vehicle detection and classification [5]. The loop detector system is based on a coil of wire placed under the roadway to capture the change in the magnetic profile signal's characteristics, such as amplitude, phase, and frequency, when a vehicle passes over it [6]. Several studies on the loop detector technique have shown its high accuracy (99% accuracy) for large vehicle classification, such as cars, trucks, and vans [7–10], it was also proven that loop detectors have no dependency on the vehicle speed [11]. Although the loop detector system is the most widely adopted in-roadway-based vehicle classification technique, it might not be the most suitable system for easy and low-cost implementation, as it requires coil installation under the roadway surface.

2.Methods

Neural networks, the main backbone for machine learning, operate in away that is analogous to biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex[32]. Neural networks use little preprocessing compared to other classification algorithms. This means that the network learns its optimal processing filters, which are manually prepared in traditional algorithms. This independence from prior knowledge and human effort feature design is a major advantage. Consequently, neural network scan efficiently find relationships between a set of input raw data(in this case, seismic wave forms) and the desired out put value(vehicle class probabilities). Neural networks consist of three main components: neurons, weights, and bias. Ina feed forward process, the neurons are determined by the values of the previous input and the weights variable that connect previous inputs to the neuron as shown in Figure1.Bias is an independent variable that acts as are fresher that perturbs the function by adding a constant. The output Yo fall neuron scan be calculated as follows:

$$Y = f\left[\sum_1^n (X \times W) + b\right]$$

Where n is the number of neurons in the previous layer, X is the value that then on holds, Wis the weight that connects Y with X, and b is the bias. The non linearity activation function f can be changed depending on the application of the neural network. To ensure a fair comparison of the three neural network models we evaluated in this study, we adopted the rectified linear unit(ReLU) [33] as an activation function after all layers. The ReLU equation returns all negative values to zero and keeps positive values:

$$f(Y) = max(0,\ Y)$$

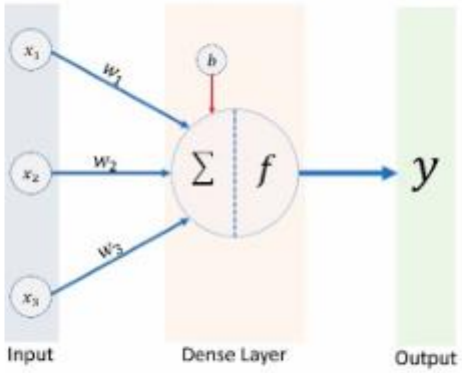


Figure 1. A simple neural network illustrating Equation (1). Inputs (x) multiplied by weights (w) are summed in the dense layer, adding bias (b), then the activation function (f) is applied to get the output. Table 1. Characteristics of the three neural network architectures used. DNN CNN RNN Number of dense layers 11 4 2 Special layer None Convolutional layer LSTM Activation function after dense layers ReLU Activation function after the final layer SoftMax Soft Max SoftMax

Figure1.AsimpleneuralnetworkillustratingEquation(1). Inputs(x)multiplied by weights(w) are summed in the dense layer, adding bias(b), then the activation function(f)is applied to get the output.

2.1. Deep Neural Network

A deep neural network (DNN) is a simple network with many hidden layers. A large number of hidden layers is advantageous for dealing with time-series data [28]. Our DNN model contains 11 hidden layers. The first four hidden layers each contain 256 neurons, the middle three layers have 128 neurons, and the last four layers have 64 neurons. This A deep neural network (DNN) is a simple network with many hidden layers. A large number of hidden layers is advantageous for dealing with time-series data [28]. Our DNN model contains 11 hidden layers. The first four hidden layers each contain 256 neurons, the middle three layers have 128 neurons, and the last four layers have 64 neurons. This decrease in neuron count helps DNN to compress the information into fewer neurons. The last layer, the output layer, contains four neurons representing the four classes in our model (Figure 2). Before each decrease in the size of the hidden layer, we apply batch normalization to avoid internal covariate shifts [35]. The details of the DNN model architecture are given in the supplementary material (Table S1). decrease in neuron count helps DNN to compress the information into fewer neurons. The last layer, the output layer, contains four neurons representing the four classes in our model (Figure 2). Before each decrease in the size of the hidden layer, we apply batch normalization to avoid internal covariate shifts [35]. The details of the DNN model architecture are given in the Supplementary Material (Table S1).

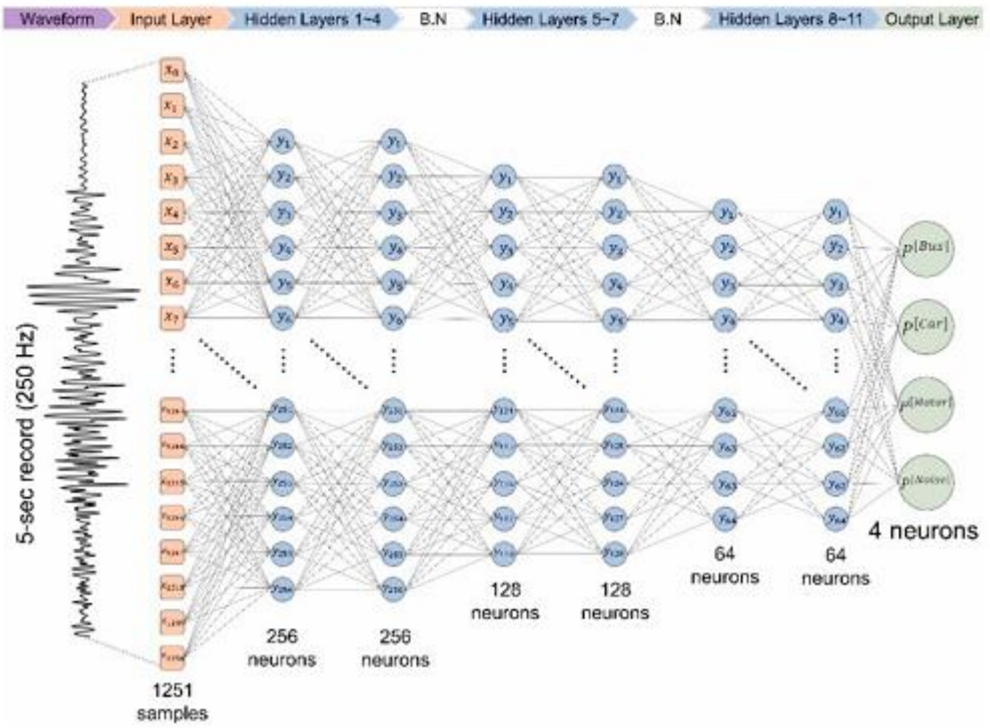


Figure 2. The DNN architecture used in this study. The 5 s waveform is discretized as 1251 samples and fed to 11 dense layers, including two batch normalization (B.N) operations between hidden Figure 2. The DNN architecture used in this study. The 5 s waveform is discretized as 1251 sam ples and fed to 11 dense layers, including two batch normalization (B.N) operations between hid den layers 4 and 5 and hidden layers 7 and 8. The model produces four values indicating the prob ability of each vehicle class. layers 4 and 5 and hidden layers 7 and 8. The model produces four values indicating the probability of each vehicle class.

2.2. Convolutional Neural Network

The convolutional neural network (CNN) has become popular for solving problems that contain features such as image recognition and is considered the best algorithm for visual recognition problems [36]. CNN contains a convolutional layer before the main neural network that is made up of multi-channel filters that extract unique features of each class. CNN thus breaks problems into smaller tasks, making the classification task for the next layers much easier [26]. The convolutional layer functions as a feature extractor, and the neural network (also called the fully connected layer) classifies based on features The convolutional layer functions as a feature extractor, and the neural network (also called the fully connected layer) classifies based on features in stead of the raw data. The CNN we used for this study contained four convolutional layers with 50 filters (sized 1 × 5) in each layer. We used Max Pool as a down sampling layer with a dimension of (1 × 3) to keep the maximum value of each of the 3 samples. So, the output of the Max Pool layer is one-third of the original data (1247/3 = 415 samples). There are 4 convolutional layers, each followed by a Max Pool layer. The final output of the convolutional layer is 50 channels signal, and each channel contains 13 features. In other words, the output is (13 × 50) the features map. We used a flatten layer to convert this map to a list with 650 variables to introduce it into the fully connected layer. The fully connected layer contains four hidden layers and a final output layer (Figure 3). The details of the CNN model architecture used in this study are listed The details of the CNN model architecture used in this study are listed in Table S2. We chose four convolutional layers after testing different numbers of layers and considering the trade-offs between accuracy and computational time. Table S2. We chose four convolutional layers after testing different numbers of layers and considering the trade-offs between accuracy and computational time.

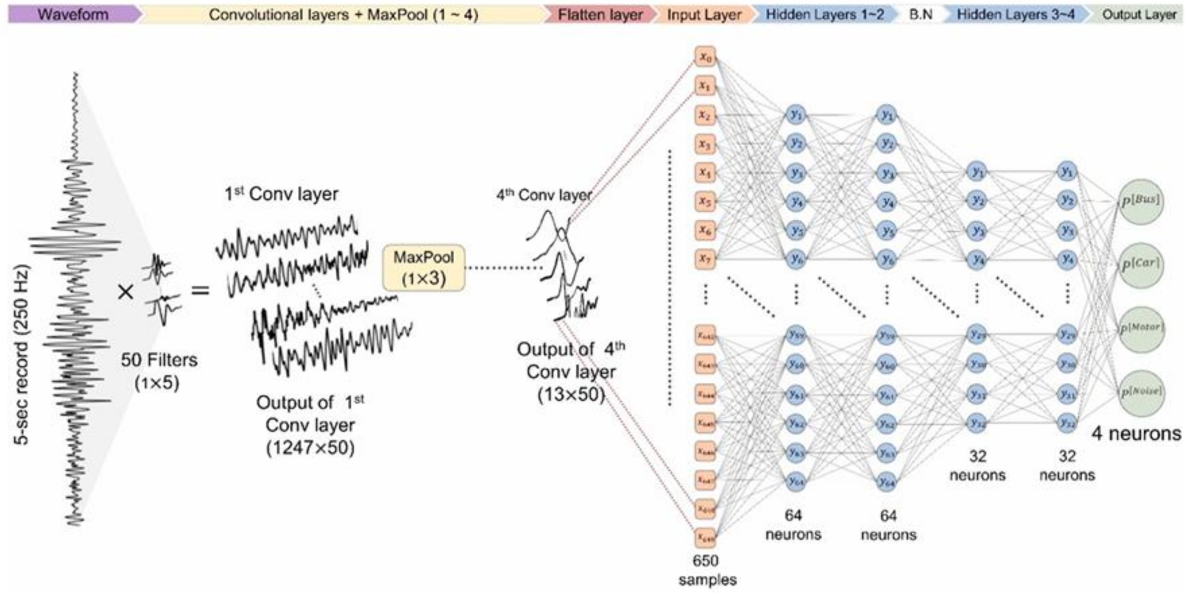


Figure 3. The CNN architecture used in this study. Five convolutional layers each contain 50 filters and a Max Pool layer to down sample the amount of contained data. The convolutional and flattening layers condense the original 1251 samples to the amount of contained data. The convolutional and flattening layers condense the original 1251 samples to 650 samples containing filtered features. These are introduced to a neural network with four hidden layers and one batch normalization (B.N) operation. The model produces four values, indicating the probability of each vehicle class. 650 samples containing filtered features. These are introduced to a neural network with four hidden layers and one batch normalization (B.N) operation. The model produces four values, indicating the probability of each vehicle class.

2.3. Recurrent Neural Network

The recurrent neural network (RNN) is a recently developed architecture in which connections between nodes form a directed graph along a temporal sequence, which allows it to exhibit temporal dynamic behavior [3]. RNN is similar to DNN, but it also includes a but it also in cludes a memory of previous results. Our RNN model used two layers of long short-term memory (LSTM) as shown in Figure 4 and Table S3 in the supplementary material. Be cause LSTM was responsible for the dramatic advancement in speech recognition [37], we anticipated a similar performance gain in seismic recognition. memory of previous results. Our RNN model used two layers of long short-term memory (LSTM) as shown in Figure 4 and Table S3 in the Supplementary Material. Because LSTM was responsible for the dramatic advancement in speech recognition [37], we anticipated a similar performance gain in seismic recognition.

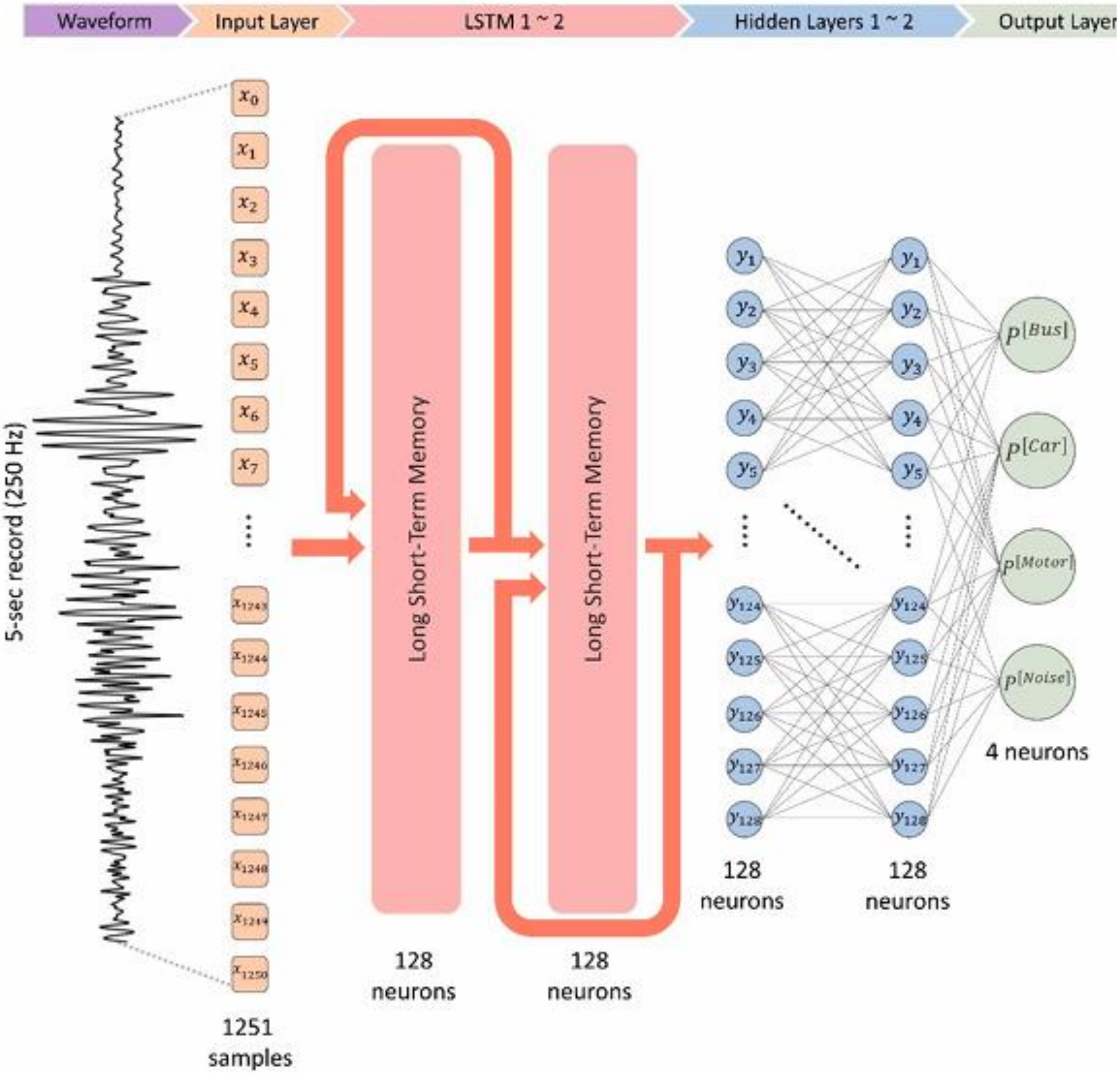


Figure 4. The RNN architecture used in this study. The model contains two LSTM layers and two hidden layers. The model produces four values, indicating the probability of each vehicle class.

hidden layers. The model produces four values, indicating the probability of each vehicle class

3. Data

3.1 Data set

In this study, we used geophones to obtain seismic data for different vehicles at Kyushu University in July 2020. We placed the geophones in three stations 15 m apart, located 0.5 m from the road. The vertical motions (vibration) were recorded at a rate of 250 Hz. We tagged vehicles by size as large (e.g., buses and trucks), medium (e.g., private passenger cars), and small (e.g., motorcycles and scooters). During the experiment, a video camera was used to provide a visual guide for the manual preparation of the training data. Each event (the passage of a vehicle) lasted 2–3 s when the vehicle was close to the geophone. Based on signals at three stations, we estimated the speeds of the vehicles. The speeds of most vehicles used in this experiment were 25~35 km/h, and the maximum speed was 45 km/h. In the training process, we chose clear vehicle signals, eliminating all signals that contained surrounding noise or that over lapped with other vehicles to avoid overfitting the models. The selected events were ex tracted from the record in the form of windows 5 s long, containing 1251 data points (5 × 250 Hz = 1250 samples). This duration was selected to guarantee the inclusion of the whole seismic waveform. We extracted, on average, 68 waveform windows per geophone station for each of the three-vehicle classes for a total of 612 windows. We also selected 318 wave form windows to represent the noise in our data as the fourth class. These include noise produced by strong winds, bicyclists, walkers, pedestrians pushing a trolley, road mainte nance, and ambient noise. These 930 windows constituted the entire input to the three neural networks; examples of each class in the dataset are shown in Figure S1 in the sup plementary material. 250 Hz =1250 samples). This duration was selected to guarantee the inclusion of the whole seismic waveform. We extracted, on average, 68 waveform windows per geophone station for each of the three-vehicle classes for a total of 612 windows. We also selected 318 waveform windows to represent the noise in our data as the fourth class. These include noise ambient noise. These 930 windows constituted the entire input to the three neural networks; examples of each class in the dataset are shown in Figure S1 in the Supplementary Material.

3.2. Training Data Augmentation

Large networks are trained using large amounts of training data to avoid overfit ting [36]. Our dataset of 930 samples was inadequate for this purpose; therefore, we generated synthetic data from our initial dataset for training purposes. We added random noise to waveforms to change their signal-to-noise ratio (SNR), as shown in Figure 5. We varied the SNR [39] from 1 to 5 as determined by the following:

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2,$$

SNR = P signal P noise = A signal 2 SNR = = , (5) A noise , (5) where P is average power and A is the root mean square amplitude. The resulting aug where P is average power and A is the root mean square amplitude. The resulting augmented dataset used for training contained 4650 synthetic samples (5 × 930).

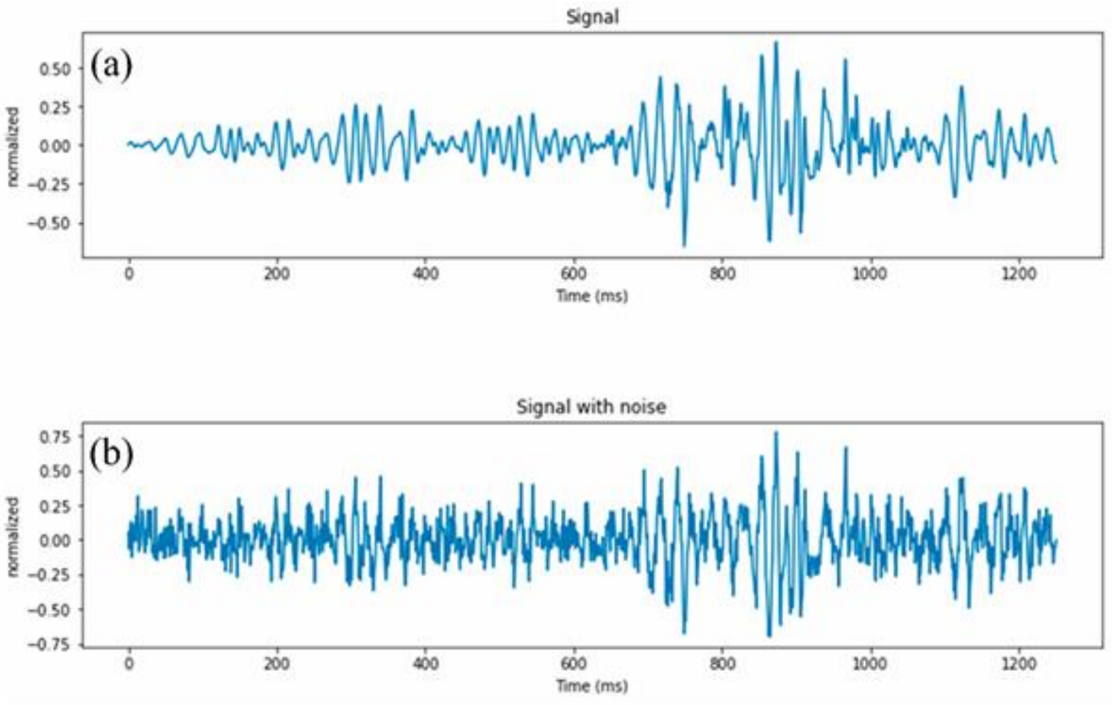
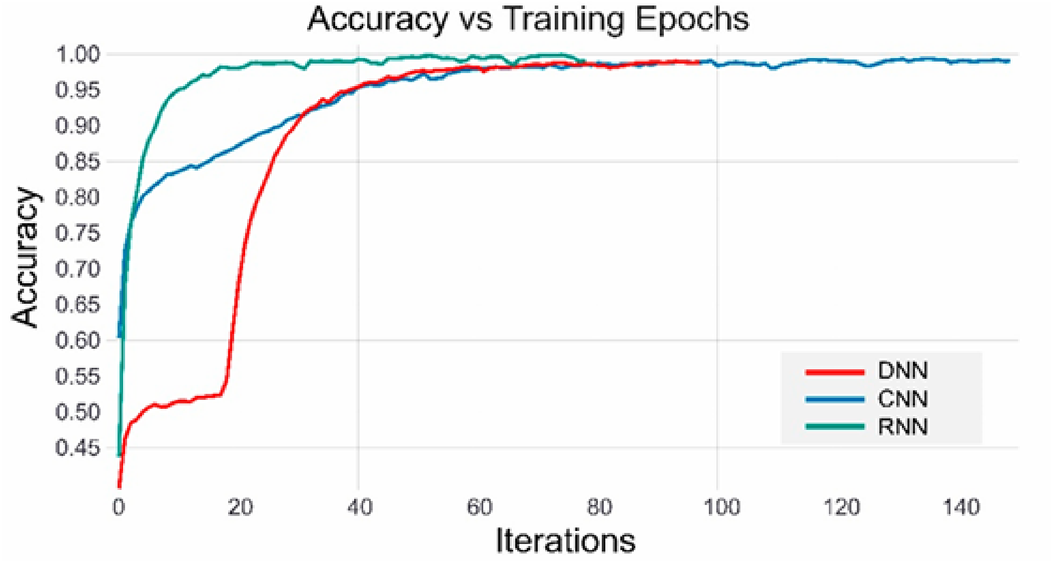


Figure 5. (a) Seismic signal of a bus before adding noise. (b) Seismic signal after adding noise to produce a signal to noise ratio of 6.

4. Results

4.1. Training and Validation

We split our augmented dataset randomly into three portions, using the scikit-learn We split our augmented dataset randomly into three portions, using the scikit-learn splitting function, dedicating 60% for training, 20% for validation, and 20% for testing. We splitting function, dedicating 60% for training, 20% for validation, and 20% for testing. We used the same training set for each of the three networks and trained them over 150 iterations, then selected the model with the best validation accuracy. We also improved our training experience and prevented overfitting in two ways. First, we applied early stopping in which the networks monitored the validation accuracy and terminated the training when accuracy did not increase for 20 iterations. Second, we set a 30% dropout chance for all weights and biases. So, in each iteration, all weights and biases have a 30% chance to be ignored in the training process. The dropout technique improves the independence of the individual weights [40]. Training took a short computation time: DNN took 87 s, CNN took 112 s, and RNN took 56 s. Because of early stopping, DNN and RNN trained for less than 150 iterations. All models showed a great improvement during training, reaching accuracies close to 99% (Figure 6).



process for DNN (red), CNN (blue), and RNN (green). RNN was stopped early at 79 iterations and DNN was stopped at 97 iterations. Figure 6. Plot showing the improvement in accuracy with increasing iterations during the training process for DNN (red), CNN (blue), and RNN (green). RNN was stopped early at 79 iterations and DNN was stopped at 97 iterations.

4.2. Classification Accuracy

We tested the classification accuracy of the three networks using 20% of the datas et (1116 samples). We compared the results with those of a similarity method for seismic event detections called template matching [41]. We randomly selected 50 waveforms for each vehicle class from the training data to be used as templates. We also recorded 15min of new data for this experiment. We took in to consideration factors that might affect the data, including the time of recording, location of stations, and types of geophones. The networks were not retrained before this exercise, and the templates also were not changed. There sulting detection accuracies are listedinTable3. DNN achieved the best accuracy, with97. 8% correct detections, followed by CNN with 96.6% and RNN with 85.3%. Template matching had much lower classification accuracy and took an order of magnitude longer to process the testing data.

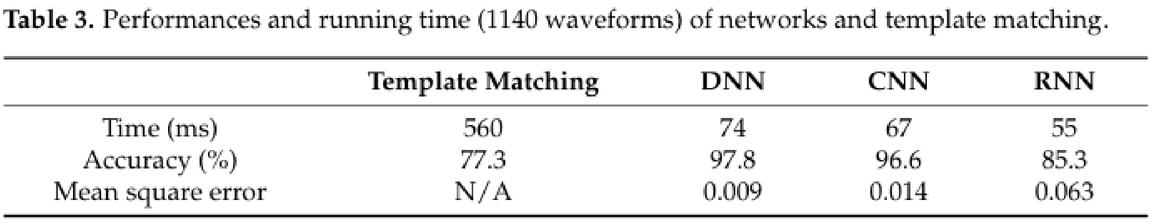


Table 3. Performances and running time (1140 waveforms) of networks and template matching.

	Template Matching	DNN	CNN	RNN
Time (ms)	560	74	67	55
Accuracy (%)	77.3	97.8	96.6	85.3
Mean square error	N/A	0.009	0.014	0.063

4.3. Vehicle Detection in Continuous Records:

Because practical applications involve records longer than 5s, we tested the framework for detecting vehicles using the 15 min continuous wave form dataset described in the previous section. The single-channel waveforms were cut in to windows 5s long, with a gap between consecutive windows of 1 stored the potential for redundant detections (Figure8). Thanks to the feature extraction implemented in the convolutional layer, CNN wasabletodetectvehiclesofdifferentclasseswithoverlappingseismicrecords. In the example of Figure 9, at ruck, a light weight car, and a motor cycle passed the geophone in quick succession. We used a 90% probability threshold to determine the predicted vehicle class. The 15 min record included 93 different vehicles. Table 4 shows the performance of the three models in terms of precision and recall per vehicle class. Precision represents the percentage of corrected clarations among all declarations made by the model, and recall represents the percentage of corrected clarations among all declarations:

$$\text{Precision}_{Class} = \frac{TP_{Class}}{TP_{Class} + FP_{Class}}$$

$$\text{Recall}_{Class} = \frac{TP_{Class}}{TP_{Class} + FN_{Class}}$$

Where TP stands for true positive, FP stands for false positive, and FN stands for false negative [42]. We used visual data, as shown in Figures 8f–hand 9 at o determinet he true positive/ true negative and ensure calculating there al accuracy for our method. By clear margins, CNN had the best precision and RNN had the best recall.

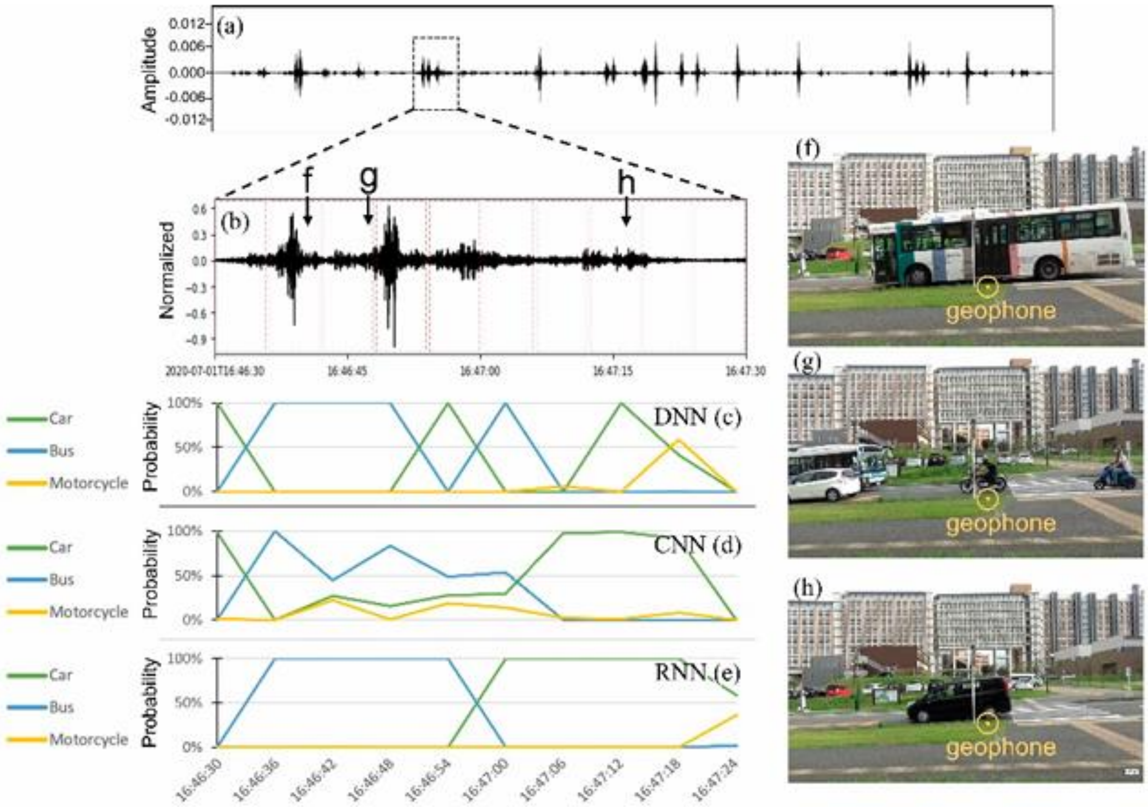


Figure 8. (a) A continuous seismic record 20 min long. (b) Detail of (a) showing a data window 1 min long divided into 5 s waveforms (red boxes) with gaps of 1 s between them. (c) The probability of vehicle types during the window in (c) is estimated using DNN, (d) CNN, and (e) RNN. Events during the window include the passage of (f) a bus, (g) a motorcycle, and (h) a car in mixed traffic. The vibrations recorded at times of pictures (f), (g,h) are displayed on panel (b).

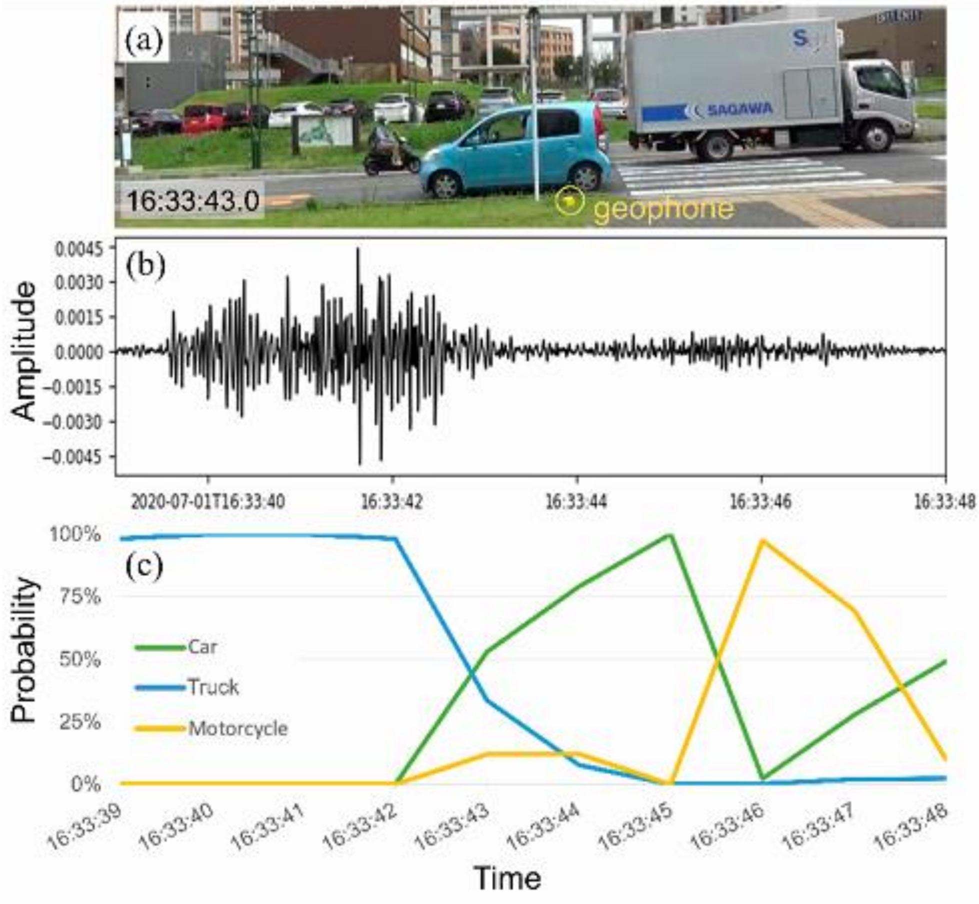
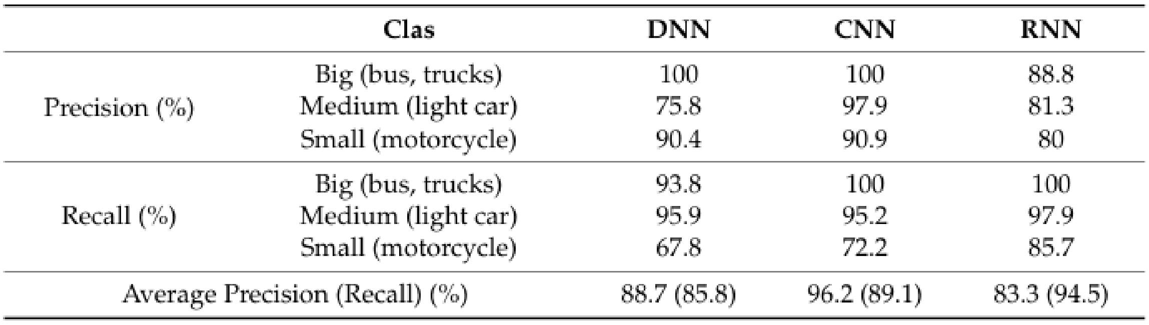


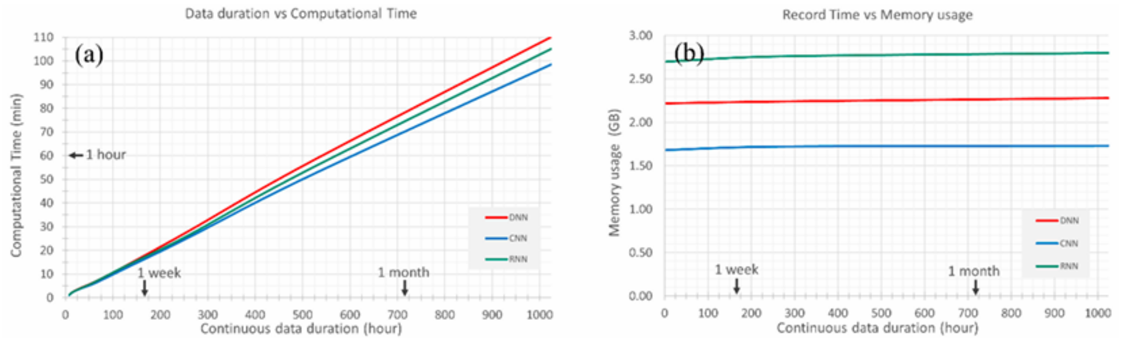
Figure 9. (a) A video frame documenting several vehicles passing the receiver at time 16:33:43. (b) The seismic noise generated a 10 s window. (c) Vehicle type probabilities estimated by CNN at intervals of 1 s during the window in (b) contain 10 interpretation points with 80% overlapping.

4.4. Scalability to Long Records

One desirable feature of a seismic-based system for traffic monitoring is its ability to operate continuously with minimal supervision, which means the system needs to deal with long records (e.g., several weeks or months). For that reason, we evaluated the computational cost of the three models, ignoring their accuracy and focusing on the scalability of networks to handle large records. We chose 1 h of data to measure running time and memory usage, then repeated the measurement safter successively doubling the size of the data set to a maximum of 1024h (nearly 43 days) (Figure10). CNN interpreted a month-long (720h) record in 70min, a computation time 10% faster than DNN. CNN also had the lowest memory usage, requiring 40% less memory than RNN. In terms of computational cost for long records, CNN was more efficient than DNN and RNN.



	Clas	DNN	CNN	RNN
Precision (%)	Big (bus, trucks)	100	100	88.8
	Medium (light car)	75.8	97.9	81.3
	Small (motorcycle)	90.4	90.9	80
Recall (%)	Big (bus, trucks)	93.8	100	100
	Medium (light car)	95.9	95.2	97.9
	Small (motorcycle)	67.8	72.2	85.7
Average Precision (Recall) (%)		88.7 (85.8)	96.2 (89.1)	83.3 (94.5)



(a)The run time required by DNN (red), CNN (blue), and RNN(green) to process aseismic record 1024h long (2.35GB).

(b) The memory usage required by the three networks to process the long seismic record, including the RAM usage and Tensor Flow in the backend.

5.Discussion

This study achieved good performance in probabilistic vehicle detection, and it confirmed the effectiveness of long-term monitoring. The neural networks out performed template matching in computation al cost and in terms of accuracy and generalization. CNN, in particular, achieved state-of-the-art performance in analyzing new data. CNN was able to detect and identify these vehicles by their frequency components, even when their signals over lapped. For example, at the time of 16:33:43 CNN determined a 40% probability for at ruckand a 60% probability for a car, even though the truck's signal was stronger than that of the car. We attributet his ability to the convolutional filters in CNN which, unlike RNN and DNN, input extracted features to the dense layers. Although RNN had the highest recall, CNN had the highest precision. Because CNN detected the over lapped vehicles with aprobability of less than 90% these identifications were not counted as detections, but the recall score could been hanced by decreasing the threshold probability to below 90%. However, CNN and other networks have failed to recognize the existence of over lapped vehicles with in the same type. The current network's architectures were not designed to count multiple vehicles. This problem could be overcome by using more than one receiver.

6. Conclusions

Machine learning proved to be an effective and low-cost technique to enable traffic monitoring in real-time based on seismic data. In this study, we evaluated three neural net work systems for this purpose and demonstrated that CNN provided the best performance in terms of accuracy and speed. CNN also surpassed the others in its ability to detect overlapping signals. RNN did not perform as well as the others for traffic monitoring because its intrinsic reliance on temporal sequences conflicts with the random nature of traffic data. Although seismic data can be used for traffic monitoring, all neural networks have a shortcoming in terms of counting vehicles because they cannot identify the presence of multiple vehicles of the same class within a waveform frame. The main limitation of neural networks is the human effort expended in acquiring and compiling a suitable amount of training data. We augmented our dataset by adding random noise. Although the models can be deployed without extra training, we recommend retraining the model as much as possible to guarantee the best performance in the generalization. Neural networks that process seismic data offer compelling advantages over current approaches to traffic monitoring. The seismic record has small file sizes compared to videos and other types of monitoring data. Because the system is simple and passive, consisting of a few geophones, it can be implemented for months at a time without supervision. The recorded data can be analyzed at a low computational cost to give clear statistical information for vehicles during the implementation period. This makes the proposed system suitable for use in hard-to-access roads. Our favored method, based on CNN, is suitable for continuous records of a month or longer; CNN was able to process a month's worth of data in approximately an hour.

References

1. Lee, H.; Coifman, B. Using LIDAR to Validate the Performance of Vehicle Classification Stations. J. Intell. Transp. Syst. Technol. Plan. Oper. 2015, 19, 355–369. [CrossRef]
2. U.S. Federal Highway Administration (Ed.) Traffic Monitoring Guide—Updated October 2016 ; Office of Highway Policy Information: Washington, DC, USA, 2016.
3. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.E.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. Heliyon 2018, 4, e00938. [CrossRef] [PubMed]
4. Won, M. Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey. IEEE Access 2019, 8, 73340–73358. [CrossRef]
5. Coifman, B.; Neelisetty, S. Improved speed estimation from single-loop detectors with high truck flow. J. Intell. Transp. Syst. Technol. Plan. Oper. 2014, 18, 138–148. [CrossRef]
6. Jeng, S.T.; Chu, L. A high-definition traffic performance monitoring system with the Inductive Loop Detector signature technology. In Proceedings of the 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014, Qingdao, China, 8–11 October 2014; Institute of Electrical and Electronics Engineers Inc.: Piscataway Township, NJ, USA, 2014; pp. 1820–1825.
7. Wu, L.; Coifman, B. Improved vehicle classification from dual-loop detectors in congested traffic. Transp. Res. Part C Emerg. Technol. 2014, 46, 222–234. [CrossRef]
8. Wu, L.; Coifman, B. Vehicle length measurement and length-based vehicle classification in congested freeway traffic. Transp. Res. Rec. 2014, 2443, 1–11. [CrossRef]

9. Balid, W.; Refai, H.H. Real-time magnetic length-based vehicle classification: Case study for inductive loops andwireless magnetometer sensors in Oklahoma state. Transp. Res. Rec. 2018, 2672, 102–111. [CrossRef]
10. Li, Y.; Tok, A.Y.C.; Ritchie, S.G. Individual Truck Speed Estimation from Advanced Single Inductive Loops. Transp. Res. Rec. 2019, 2673, 272–284. [CrossRef]
11. Lamas-Seco, J.J.; Castro, P.M.; Dapena, A.; Vazquez-Araujo, F.J. Vehicle classification using the discrete fourier transform with traffic inductive sensors. Sensors 2015, 15, 27201–27214. [CrossRef] [PubMed]
12. Odat, E.; Shamma, J.S.; Claudel, C. Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors. IEEE Trans. Intell. Transp. Syst. 2018, 19, 1593–1606. [CrossRef]
13. Dong, H.; Wang, X.; Zhang, C.; He, R.; Jia, L.; Qin, Y. Improved Robust Vehicle Detection and Identification Based on Single Magnetic Sensor. IEEE Access 2018, 6, 5247–5255. [CrossRef]
14. Belenguer, F.M.; Martinez-Millana, A.; Salcedo, A.M.; Nunez, J.H.A. Vehicle Identification by Means of Radio-Frequency Identification Cards and Magnetic Loops. IEEE Trans. Intell. Transp. Syst. 2019, 21, 5051–5059. [CrossRef]
15. Li, F.; Lv, Z. Reliable vehicle type recognition based on information fusion in multiple sensor networks. Comput. Netw. 2017, 117, 76–84. [CrossRef]