sensors

MDPI

Article

Vehicular Environment Identification Based on Channel State Information and Deep Learning

Soheyb Ribouh [1], Rahmad Sadli [2], Yassin Elhillali [2], Atika Rivenq [2] and Abdenour Hadid [3,*]

[1] Normandie Univ. Rouen, LITIS (Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes), Av. de l'Université le Madrillet, 76801 Saint Etienne du Rouvray, France; soheyb.ribouh@univ-rouen.fr (S.R.)
[2] Univ. Polytechnique Hauts-de-France, UMR 8520, IEMN, Institut d'Électronique de Microélectronique et de Nanotechnologie, DOAE, Département d'Opto-Acousto-Électronique; 59300, Valenciennes, France rahmad.sadli@uphf.fr (R.S.); yassin.elhillali@uphf.fr (Y.E.); atika.menhaj@uphf.fr (A.R.)
[3] Sorbonne Center for Artificial Intelligence, Sorbonne University Abu Dhabi, Abu Dhabi, United Arab Emirates
* Correspondence: abdenour.hadid@sorbonne.ae (A.H.)

Abstract: This paper presents a novel vehicular environment identification approach based on deep learning. It consists of exploiting the vehicular wireless channel characteristics in the form of Channel State Information (CSI) in the receiver side of a connected vehicle in order to identify the environment type in which the vehicle is driving, without any need to implement specific sensors such as cameras or radars. We consider environment identification as a classification problem, and propose a new convolutional neural network (CNN) architecture to deal with it. The estimated CSI is used as the input feature to train the model. To perform the identification process, the model is targeted for implementation in an autonomous vehicle connected to a vehicular network (VN). The proposed model is extensively evaluated, showing that it can reliably recognize the surrounding environment with high accuracy (96.48%). Our model is compared to related approaches and state-of-the-art classification architectures. The experiments show that our proposed model yields favorable performance compared to all other considered methods.

Keywords: Vehicle-To-Everything (V2X) communications; channel state information; deep learning; vehicular network; autonomous vehicle; intelligent transportation systems

check for updates

Citation: Ribouh, S.; Sadli, R.; Elhillali, Y.; Rivenq, A.; Hadid, A. Vehicular Environment Identification Based on Channel State Information and Deep Learning. *Sensors* **2022**, *1*, 0. https://doi.org/

Academic Editor: Yichuang Sun

Received: 21 October 2022
Accepted: 19 November 2022
Published:

1. Introduction

Autonomous connected vehicles have been the focus of recent research works on intelligent transportation systems (ITS), in which autonomous vehicles are anticipated to be widely used as part of the smart road vision and the next generation of transportation systems. The development of autonomous driving system aims to achieve the highest level of autonomy, at which no driver is required. When this goal is met, Vehicle-To-Everything (V2X) communications will emerge as a paramount enabler for leveraging the full potential of these vehicles. Furthermore, V2X communication is mandatory to ensure the transition from self-autonomy to full collaborative autonomy [1–3]. Thus, to allow connectivity between vehicles, vehicular networks (VN) should be set up in ad hoc fashion by forming Vehicle Ad Hoc Networks (VANETs) and Mobile Ad Hoc Networks (MANETs) [4]. Because V2X communication is quite important, the automotive industry is declaring its intent to deploy V2X communication technology in their future cars. Moreover, it is further supported by transportation system governments, such as the proposed mandate from the National Highway Traffic and Safety Administration (NHTSA) that suggests all vehicles have V2X capability [?]. On the other hand, the goal of deploying autonomous vehicles is to improve road safety through cooperative driving that uses the available roadway efficiently and reduces road congestion.

According to the NHTSA, most crash accidents are caused by vehicles traveling over the speed limit. Consequently, in order to provide road safety, autonomous vehicles should be aware of the speed limit and the environment. Thus identifying the type of environment in which the vehicle is driving allows the vehicle to make good a self-decision as to the correct driving speed.

In this context, Artificial Intelligence (AI) has been established as a leading actor towards the developments of intelligent systems, enabling autonomous vehicles to make correct decisions [6,7]. Thus, in this paper we introduce a new approach towards vehicular environment identification without the need for specific sensors. The proposed method consists of using exchanged Cooperative Awareness Messages (CAM) between vehicles as well as between vehicles and infrastructure to explore channel characteristics, which are then used to recognize the vehicular environment, as shown in Figure 1.

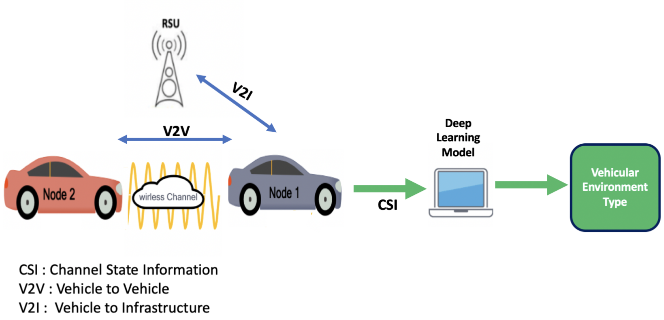

RSU
V2I
V2V
Node 2
wirless Channel
Node 1
CSI
Deep Learning Model
Vehicular Environment Type
CSI : Channel State Information
V2V : Vehicle to Vehicle
V2I : Vehicle to Infrastructure

Figure 1. Vehicular environment identification process.

2. Related Work

In the literature, many research works have focused on deep learning-based environment perception in order to make critical decisions such as vehicle speed in the context of correct decision-making for autonomous cars. In [8], the authors proposed a new method called the integrated perception approach to construct the environment. They used road information data such as the distances to surrounding lane markings provided from video images. These data were used as the input features of a neural network model in order to reach the correct driving decisions.

A highway environment identification approach has been presented in [9]. The authors used video data of a highway area recorded under various weather conditions in order to develop a vision system for recognizing the bounds of highway areas and updating the vehicle with respect to the highway driving conditions. In [10], the authors presented a new perception method for urban environments. Their approach was based on the use of video images provided from an embedded camera in a vehicle, which were then used to train a neural network in order to develop a conditional navigation model that allows for prior reception of high-level directional commands.

A vehicular urban environment perception method for autonomous vehicles was presented in [11]. The approach consists of a Global Positioning System (GPS), Radar, and Light Detection And Ranging (LiDAR)-based data fusion algorithm for reaching safe driving decisions. An environment perception approach for a self-driving vehicle in an urban area was established in [6]. In this method, the authors used a 64-beam rotating LiDAR with a specific unsupervised algorithm, then generated high-resolution maps of the surrounding environment, allowing the vehicle to enable the suitable driving parameters for its environment. The authors of [12] established an approach based on the use of data fusion in order to obtain a presentation of the environment that includes a camera, 360-degree LiDAR, and GPS/Inertial Measurement Unit (IMU) sensors deployed in a vehicle. Thus, the vehicle can make correct self-driving decisions depending on the environment in which it drives. In [13], the authors proposed an environment perception framework to enhance the environmental awareness of autonomous vehicles. This framework incorporates Voxel Region-based Convolution Neural Network (PVRCNN)-based vision features and leverages Vehicle-to-Infrastructure (V2I) communication technology. The Normal Distributions Transform (NDT) point cloud registration algorithm is used both onboard and at the roadside to obtain the position of autonomous vehicles and objects detected by the multi-sensor system at the roadside are sent back to the autonomous vehicles to improve their perception. An end-to-end machine learning model that combines control algorithms, convolutional neural networks (CNNs), and multitask (MT) learning for autonomous driving was introduced in [14]. The proposed model is able to simultaneously perform regression and classification tasks for estimating perception indicators and driving decisions, and can be used to evaluate inference efficiency and driving stability. In [15], a new approach of enhancement perception for Autonomous Driving Using Semantic and Geometric Data Fusion was presented based on low-level fusion of semantic scene information and geometry from LiDAR-based 3D point clouds. This method provides better range coverage and enables improved perception through 3D object classification and detection. In [16], the authors introduced real-time object identification, distance estimation, and instantaneous position tracking in all environmental conditions using a deep learning algorithm with no additional sensors. The proposed framework was implemented on a Raspberry Pi 4 Model B using the Raspberry Pi NoIR Camera Module V2.

Almost all of the approaches described above are essentially based on the use of specific sensors such as cameras, radars, and LIDARs. Data collection based on these sensors requires a significant amount of computing resources and power [17].

To avoid this, we propose a novel environment identification approach based on deep learning dedicated to autonomous vehicles without the need for specific sensors. For this, we exploit the shared wireless channel characteristics between vehicles communicating in vehicular networks.

Because the CSI values are the most accurate representation of wireless channel characteristics [18], we use the CSI values estimated from the packets exchanged between vehicles through Vehicle-to-Vehicle (V2V) communications as input features for our proposed convolutional neural network model. This model is able to reliably identify the surrounding environment by learning the channel characteristics (CSI) for each environment. Thus, the vehicle can set up the right automotive driving parameters (such as speed limits) corresponding to the identified environment.

The remainder of this paper is organized as follows. Section 3 describes our wireless communication model. Section 4 provides an overview of the proposed vehicular environment identification process, while Section 5 describes our tests setups and the evaluation of the performance of our proposed method. Finally, we provide our conclusions in Section 6.

3. System Model

To begin, we establish a wireless communication vehicular network model in which each vehicle uses a half-duplex transmitter/receiver pair to communicate with other vehicles. These vehicles exploit the wireless channel effect (characterized by CSI) on the received messages as the input features for the CNN model used to identify the vehicular environment.

The proposed V2X network operates on the IEEE 802.11p standard. The main physical (PHY) layer of this protocol is based on the Orthogonal Frequency Division Multiplex (OFDM) waveform. The exchanged frames in the vehicular network are constituted as shown in the figure below (Figure 2).

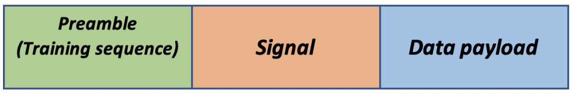


Figure 2. IEEE 802.11p PHY layer frame structure.

The vehicular wireless channel is structured as a double selective fading propagation channel, which is characterized by the delay spread and the Doppler spread [18]. The base-band time-varying response of the multi-path channel is provided by

$$\begin{aligned} h(t,\tau) &= \sum_{l=0}^{L-1} A_l(t,\tau)\delta(\tau-\tau_l(t)), Where \\ A_l(t,\tau) &= |A_l(t,\tau)| \exp[j(2\pi f_0 \tau_l(t) + \phi_l(t,\tau)] \end{aligned} \tag{1}$$

where L represents the number of non-zero paths, $A_l(t)$ represents the time-varying complex amplitudes, and $\tau_l(t)$ represents the time-varying path delays. Moreover, note that the phase of the complex amplitude $A_l(t)$ in this instance depends on the variation of Doppler shift. In addition to the time delay, the signal's transmission over this channel may cause a Doppler shift in each path. As a result, the various delayed and frequency-shifted versions of the transmitted signal are superimposed at the receiver side [19].

We assume that the channel characteristics are static over a constant time T_c (coherence time) [20], which is inversely proportional to the maximum Doppler shift f_d:

$$T_c \approx \frac{0.423}{f_d} \tag{2}$$

In vehicular communication, f_d can be expressed by the speed difference between the two communicating vehicles ΔV, as shown below:

$$\begin{aligned} f_d &= \frac{\Delta V}{c} f_0 \\ \Delta V &= |V_1 - V_2| \end{aligned} \tag{3}$$

where c and f_0 represent the celerity (speed of light) and the communication center frequency, respectively.

Depending on the coherence bandwidth ($1/T_c$), when the channel's coherence bandwidth exceeds the signal's bandwidth, the channel exhibits flat fading. When the coherence bandwidth of the channel is smaller than the bandwidth of the signal, it is known as a frequency-selective fading channel (inter-symbol interference in the time domain).

According to the European Telecommunications Standards Institute (ETSI), the V2X scenario has a major impact on wave propagation, and thus the channel model [21]. We can consider five major vehicular environments depending on the different channel modeling characteristics of power, delay, and doppler [19,22,23]. These vehicular environment characteristics are shown in Table 1.

Table 1. Vehicular environment characteristics.

	Taps	Power [dB]	Delay [ns]	Doppler [Hz]
U-LOS	**Tap 1**	0	0	0
	Tap 2	−8	117	236
	Tap 3	−10	183	−157
	Tap 4	−15	333	492
U-NLOS	**Tap 1**	0	0	0
	Tap 2	−3	267	295
	Tap 3	−4	400	−98
	Tap 4	−10	533	591
R-LOS	**Tap 1**	0	0	0
	Tap 2	−14	83	492
	Tap 3	−17	183	−295
H-LOS	**Tap 1**	0	0	0
	Tap 2	−10	100	689
	Tap 3	−15	167	−492
	Tap 4	−20	500	886
H-NLOS	**Tap 1**	0	0	0
	Tap 2	−2	200	689
	Tap 3	−5	433	−492
	Tap 4	−7	700	886

Because the vehicular environment is highly mobile, the transmitted messages are affected by the wireless channel. The received signal over the vehicular wireless channel can be written as

$$Y(k) = X(k)H(k) + W(k) \tag{4}$$

where $X(k)$ denotes the transmitted data symbols, W is the noise in the receiver, and $H(k)$ denotes the wireless channel response. This channel response is characterized by the CSI. At the receiver side, a channel estimation task is mandatory; this aims to calculate the CSI, which is required in order to recover the transmitted data. Because channel estimation is quite important in V2X communications, a great deal of research work has been carried out in this field [24–26]. The channel estimation approaches in the literature are mainly based on observation and long training sequences (LTS). The most common channel estimation method used in industrial implementations on V2X communication boards is the LS (Least Square) estimator, thanks to its low complexity; it can be expressed as

$$\hat{H}_{LS} = \underset{H_{LS}}{min} ||Y_t - X_t.H||_2^2 \tag{5}$$

where $||.||_2$ is the L_2 norm, X_t is the long training sequence vector, and Y_t denotes the corresponding observation vector. A close optimization of the LS estimator was established in [27], as follows:

$$\hat{H}_{LS} = X_t^{-1} Y_t \tag{6}$$

4. Vehicular Environment Identification Methodology

In this work, we consider the vehicular environment identification process as a multi-class classification problem. Thus, we propose two methods to tackle it, as shown in Figure 3. The first is based on the use of the long training sequences (LTSs) of the received frame, while the second approach is based on the calculated CSI values. Both the LTSs and the CSI values include 128 data samples, and these samples are used as the input features of the CNN model.

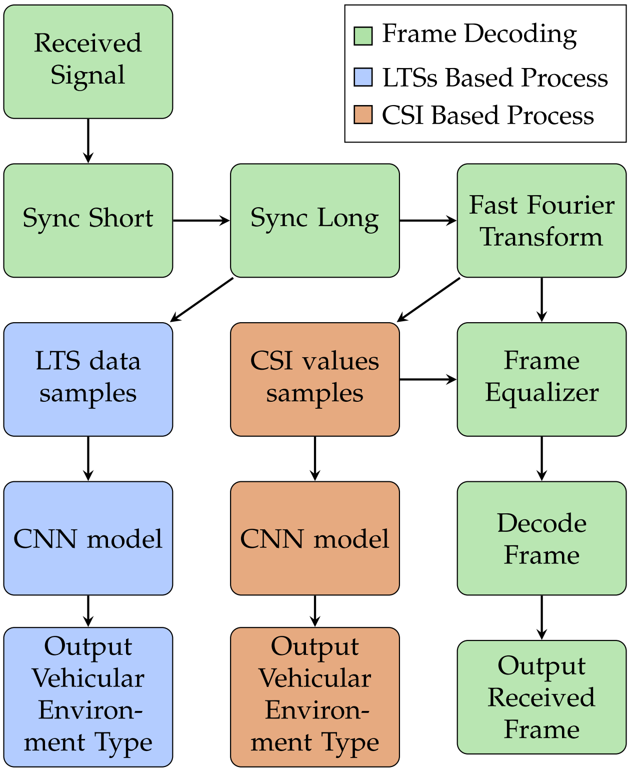


Figure 3. Flow chart describing vehicular environment identification process.

4.1. The Proposed Model

To tackle to problem of vehicular environment identification, we propose the Convolutional Neural Network (CNN) architecture shown in Figure 4.

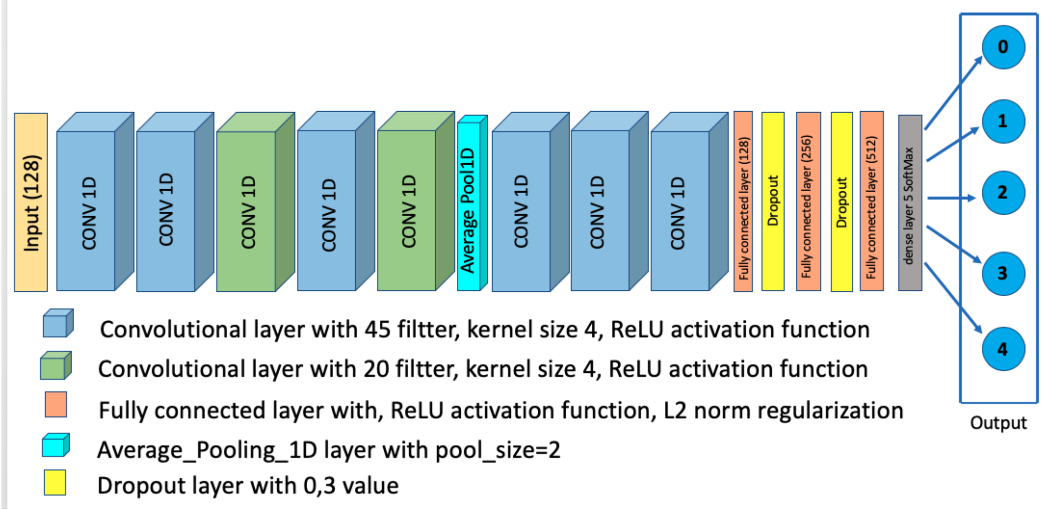

Input (128)
CONV 1D
CONV 1D
CONV 1D
CONV 1D
CONV 1D
Average Pool1D
CONV 1D
CONV 1D
CONV 1D
Fully connected layer (128)
Dropout
Fully connected layer (256)
Dropout
Fully connected layer (512)
dense layer 5 SoftMax
0
1
2
3
4
Output
Convolutional layer with 45 filtter, kernel size 4, ReLU activation function
Convolutional layer with 20 filtter, kernel size 4, ReLU activation function
Fully connected layer with, ReLU activation function, L2 norm regularization
Average_Pooling_1D layer with pool_size=2
Dropout layer with 0,3 value

Figure 4. Proposed CNN Architecture.

The proposed CNN model is constructed as follows: First, it begins with two similar one-dimensional (1D) convolutional layers; these two layers include 45 filters. Then, we have a third 1D convolutional layer, including 20 filters. This is followed by two other 1D convolutional layers that include 45 and 20 filters, respectively. The size of the filters utilized in all the previous convolutional layers is $(4X1)$. After that, we have an average pooling layer with a pool size of 2. This average pooling layer has three 1D convolutional layers and employs 45 fillers of $(4X1)$ kernel size. The ReLu function is used as an activation layer for all the previous convolutional layers. These layers are followed by three fully connected layers, which include 128, 256, and 512 neurons respectively, with the ReLu activation function used for these three dense layers. To reduce the overfitting effect, we add two dropout layers (with $p = 0,3$) after the first and the second fully connected layer. Furthermore, second-norm regularization is used for all the fully connected layers [28]. Finally, the output layer is a fully connected layer in which the number of neurons is 5 (equal to the number of environment classes), with SoftMax used as the activation function.

The proposed architecture was built using the Tensorflow library [29], and we used 20 epochs and a batch size of 50 to train the model.

4.2. Data-Set Generation

For training, we considered 5 classes of vehicular environments: Rural LOS (Line-of-sight), Urban LOS, Urban NLOS (Non-Line-of-sight), Highway LOS, and Highway NLOS. Each environment is modeled by a wireless channel based on real-world vehicular environment measurement of the delay, gain, and Doppler frequency.

The vehicular channel characteristics of each environment can be found in Table 1. A label is assigned to each environment corresponding to the class outputs of the CNN model (Table 2).

Table 2. Vehicular environment labels and required speed limits.

Vehicular Environment	Label	Speed Limits
Highway NLOS	0	130 km/h
Highway LOS	1	130 km/h
Rural LOS	2	90 km/h
Urban LOS	3	50 km/h
Urban NLOS	4	50 km/h

To generate the dataset samples we employ a half-duplex V2V communication based on OFDM, which was developed using Matlab. To simulate the different vehicular environments (wireless channels models) we used the V2VChannel framework in Matlab, which is referenced in [19].

Several 802.11p packets are transmitted through the different channel models. For each environment, the packets are transmitted at a different value of the Signal-to-Noise Ratio (SNR), where the SNR range is from 15 dB to 40 dB with a step of 0.5 dB.

This process was repeated 400 times with different releases of the channel model for each environment. At each step, we computed the LTS in the received packet and the calculated CSI values, obtaining the 128 symbols (features) of both LTS and CSI associated with the specific label corresponding to each environment, as shown in Table 2.

The saved sequence features (F_i) for either CSI or LTS can be expressed as

$$F_i = [[A(1), A(2), ..., A(N)]] \tag{7}$$

where $A(i)$ is the CSI or LTS sample. At the end of the process, we had 100,000 dataset samples, of which we used 80% as the training set and 20% as the validation set.

5. Evaluation and Results

In order to assess the validity and accuracy of the proposed model, we evaluated the system on different datasets. We generated a test set by transmitting several 802.11p packets through the different channel models (V2VChannel framework of Matlab). For each environment, the SNR range was set from 15 dB to 40 dB with a step of 0.25 dB. This process was repeated 30 times with different releases of the channel model for each environment, resulting in 15,000 test sequences (LTS and CSI).

Before evaluating the proposed architecture, we trained our model using the categorical cross-entropy loss function and the Adam optimizer [30]. The training process was carried out on a machine including an NVIDIA Tesla P100 GPU.

Because both the CSI and LTS are complex numbers, we use training and test sets with three configurations depending on the input data format for each configuration. We use the magnitude for the first test-bed, the angle for the second, and a two-channel input for the third one, wherein the real part of the complex number is used for the first-channel input and the imaginary part is set for the second-channel input.

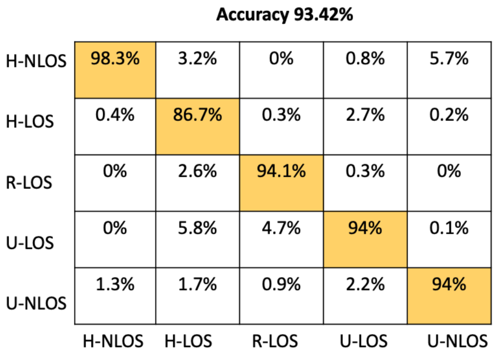
5.1. LTS Approach Performance Evaluation

As shown in Table 3, the two-channel configurations has high accuracy, achieving 93.42%, which is better than the magnitude and the angle configurations, which are 92.22% and 91.78% respectively.

Table 3. LTS approach accuracy for magnitude angle and two-channel configurations.

Configuration	Accuracy
Magnitude	92.22 %
Angle	91.78 %
2-Channel	93.42 %

Figure 5 represents the confusion matrix of the test samples for the proposed CNN architecture using the LTS as input features within a two-channel configuration. From this confusion matrix, it can be seen that our proposed CNN model is able to reliably recognize the different vehicular environment; it correctly identifies the H-NLOS and H-LOS environments with an individual accuracy of 98.3% and 86.7%, respectively, and the R-LOS, U-LOS, and U-NLOS environments with 94% accuracy.



Accuracy 93.42%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	98.3%	3.2%	0%	0.8%	5.7%
H-LOS	0.4%	86.7%	0.3%	2.7%	0.2%
R-LOS	0%	2.6%	94.1%	0.3%	0%
U-LOS	0%	5.8%	4.7%	94%	0.1%
U-NLOS	1.3%	1.7%	0.9%	2.2%	94%

Figure 5. Confusion matrix based on LTS approach for the proposed CNN.

We compared the proposed CNN architecture to an ANN architecture containing four dense fully connected layers, including 64, 128, 256, and 512 neurons before the output layer, each of which have five neurons (equal to the number of environments to identify). Other machine learning classifier candidates used for comparison are the Random Forest classifier (RF, with 100 trees), K-Neighbors classifier (K-NN), where K was set to five neighbors, Gaussian Naive Bayes (GNB), and Support Vector Machine (SVM) with a linear kernel.

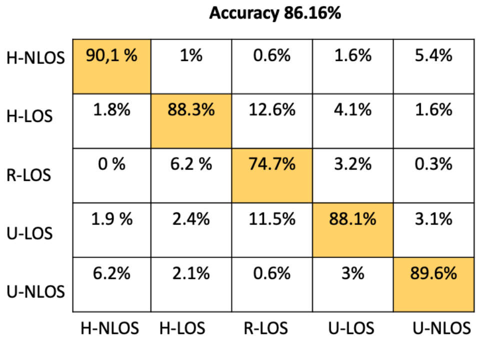
Table 4 shows the comparison between our proposed model and the approaches mentioned above based on test accuracy and environment identification prediction time. The prediction time has been determined using an NVIDIA Tesla P100 GPU. From Table 4, it is obvious that the prediction time of our proposed CNN Architecture has better performance than either SVM or K-NN, providing a prediction time of 51.33 µs. This time is comparable to the other approaches (ANN, RF, GNB) that have llowess prediction times; moreover, the accuracy of our model is significantly greater than these approaches; indeed, it has the best overall test accuracy at 93.42%.

Table 4. Classification accuracy and average prediction time comparison for LTS approach.

Approach	Accuracy (%)	Prediction Time (µs)
Proposed CNN	93.42	51.33
ANN	86.16	23.11
RF	68.34	25.71
K-NN	63.18	7180
GBN	20.62	4.11
SVM	31.38	10499

In order to provide more detail about the test accuracy classification, the confusion matrices of all the considered approaches are presented in Figures 5–10.

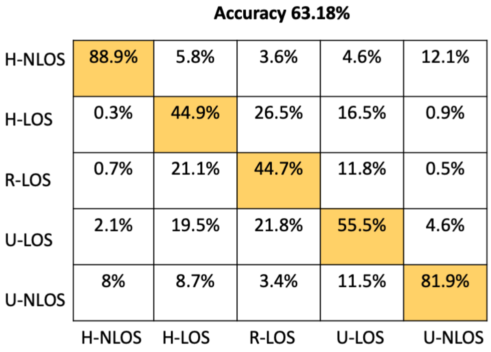
It can be seen that the K-NN and RF approaches can identify H-NLOS and U-LOS environments with acceptable individual accuracy (up to 80%) and provide less than 65% of individual test accuracy for H-LOS, R-LOS, and U-LOS environments. From Figures 9 and 10, it is clear that both the GNB and SVM classifiers fail to provide reliable environment identification.



Accuracy 86.16%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	90,1 %	1%	0.6%	1.6%	5.4%
H-LOS	1.8%	88.3%	12.6%	4.1%	1.6%
R-LOS	0 %	6.2 %	74.7%	3.2%	0.3%
U-LOS	1.9 %	2.4%	11.5%	88.1%	3.1%
U-NLOS	6.2%	2.1%	0.6%	3%	89.6%

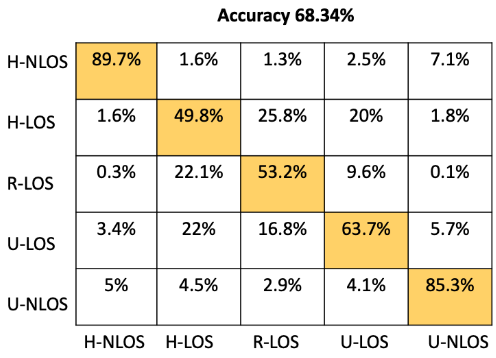
Figure 6. Confusion matrix for ANN based on LTS approach.



Accuracy 63.18%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	88.9%	5.8%	3.6%	4.6%	12.1%
H-LOS	0.3%	44.9%	26.5%	16.5%	0.9%
R-LOS	0.7%	21.1%	44.7%	11.8%	0.5%
U-LOS	2.1%	19.5%	21.8%	55.5%	4.6%
U-NLOS	8%	8.7%	3.4%	11.5%	81.9%

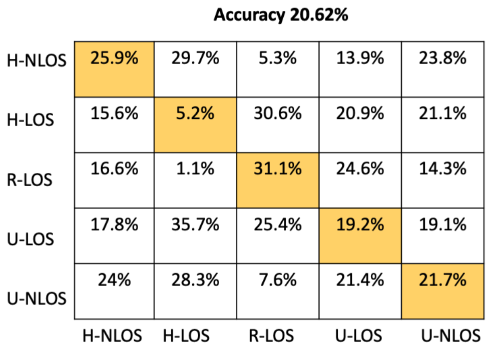
Figure 7. Confusion matrix for KNN based on LTS approach.



Accuracy 68.34%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	89.7%	1.6%	1.3%	2.5%	7.1%
H-LOS	1.6%	49.8%	25.8%	20%	1.8%
R-LOS	0.3%	22.1%	53.2%	9.6%	0.1%
U-LOS	3.4%	22%	16.8%	63.7%	5.7%
U-NLOS	5%	4.5%	2.9%	4.1%	85.3%

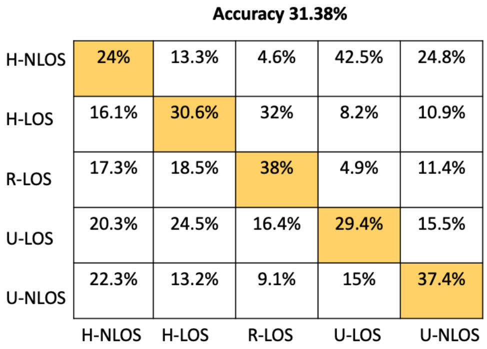
Figure 8. Confusion matrix for RF based on LTS approach.



Accuracy 20.62%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	25.9%	29.7%	5.3%	13.9%	23.8%
H-LOS	15.6%	5.2%	30.6%	20.9%	21.1%
R-LOS	16.6%	1.1%	31.1%	24.6%	14.3%
U-LOS	17.8%	35.7%	25.4%	19.2%	19.1%
U-NLOS	24%	28.3%	7.6%	21.4%	21.7%

Figure 9. Confusion matrix for GNB based on LTS approach.



Accuracy 31.38%

	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS
H-NLOS	24%	13.3%	4.6%	42.5%	24.8%
H-LOS	16.1%	30.6%	32%	8.2%	10.9%
R-LOS	17.3%	18.5%	38%	4.9%	11.4%
U-LOS	20.3%	24.5%	16.4%	29.4%	15.5%
U-NLOS	22.3%	13.2%	9.1%	15%	37.4%

Figure 10. Confusion matrix for SVM based on LTS approach.

5.2. CSI Approach Performance Evaluation

Vehicular environment identification based on the CSI approach was performed using three input feature configuration: two-channel, magnitude, and angle.

From Table 5, it is clear that that the two-channel input feature configuration provides the best performances; it has 96.48% accuracy, which is greater than the accuracy provided by the LTS approach (93.42% in two-channel configuration).

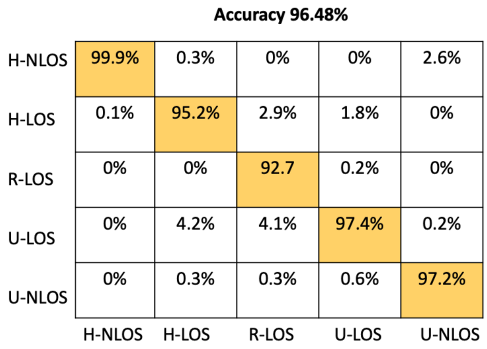
Table 5. CSI approach accuracy for magnitude angle and two-channel configurations.

Configuration	**Accuracy**
Magnitude	90.63 %
Angle	91.50 %
2-Channel	96.48 %

Figure 11 presents the confusion matrix of the proposed CNN architecture on the test set when using CSI values as input features for the model. The presented results are calculated taking into account a two-channel input shape, as this provides the most accurate performance. From this confusion matrix, it is clearly apparent that our proposed model can reliably identify all the vehicular environments based on CSI values, with high individual accuracy up to 92%.

Our model achieves 99.9%, 95.2%, 92.7%, 97.4%, and 97.2% for H-NLOS, H-LOS, R-LOS, U-LOS, and U-LOS environments, respectively.

The proposed CNN model based on CSI was compared to the related machine learning classifiers RF, K-NN, GBN, and SVM, as well as to an ANN architecture, in terms of test accuracy and the average time required to identify the environment (prediction time, performed using an NVIDIA Tesla P100 GPU). The ANN architecture and parameter settings of these classifiers are the same as described in Section 5.1.



Accuracy 96.48%

H-NLOS	99.9%	0.3%	0%	0%	2.6%
H-LOS	0.1%	95.2%	2.9%	1.8%	0%
R-LOS	0%	0%	92.7	0.2%	0%
U-LOS	0%	4.2%	4.1%	97.4%	0.2%
U-NLOS	0%	0.3%	0.3%	0.6%	97.2%
	H-NLOS	H-LOS	R-LOS	U-LOS	U-NLOS

Figure 11. Confusion matrix for the proposed CNN based on CSI approach.

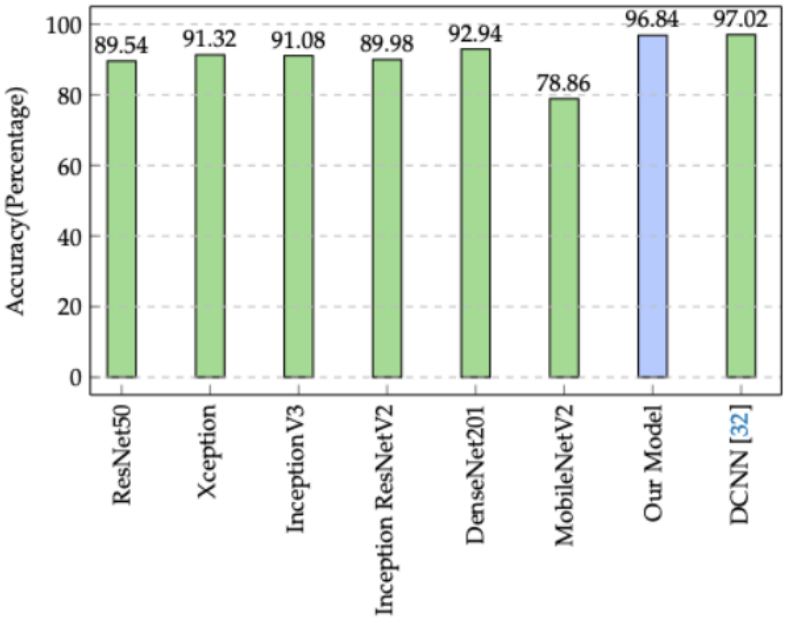


Figure 12. Comparison of our model's accuracy to state-of-the-art alternatives.

From Table 6, it can be seen that our proposed CNN model has better performance than either SVM or K-NN in terms of prediction time, as it yields in 39.56 µs. While this achieved prediction time is comparable to the other approaches (ANN, RF, GNB) that have low prediction time, in term of the test accuracy our CNN model highly outperforms all the other approaches, reaching 96.48%.

Table 6. Classification accuracy and average prediction time comparison for CSI approach.

Approach	Accuracy (%)	Prediction Time (µs)
Proposed CNN	96.48	39.56
ANN	85.64	21.11
RF	67.77	24.04
K-NN	59.26	8999
GNB	27.06	4.38
SVM	32.33	15756

According to these performance results, it is clear that the CSI approach is more accurate than the LTS approach in terms of both test accuracy and environment prediction time.

5.3. Comparison between Our Model and State-of-the-Art Architectures

Our proposed CNN architecture was compared against the popular state-of-the-art classification architectures ResNet50 [31], Xception [32], InceptionV3 [33], InceptionResNetV2 [34], DenseNet201 [35], and MobileNetV2 [36]. We trained these architectures on the same training datasets. Then, we evaluated their classification performances on the test set. Prior to this process, we updated the input shape of the input layer to fit our data inputs and updated the output layer size to five classes in order to equal the number of vehicular environments to identify.

Because the previously mentioned state-of-the-art architectures are designed to receive two-dimensional (2D) inputs for shape size, we considered a 2D channel matrix in the input features instead of a 1D channel vector. Thus, we rearranged our dataset from 1D to 2D, as follows:

$$\underset{[n*n]}{H_{2D}} = Diag(\underset{[1*n]}{H_{1D}}) \tag{8}$$

where $Diag()$ is the diagonal matrix, H_{2D} and H_{1D} are the channel matrix and corresponding channel vector, respectively, where their coefficients are the CSI values, and n is equal to 128, which is equivalent to the number of CSI values estimated per packet; thus, we have an input shape size of 128 × 128.

In Table 7, our proposed model is compared to the indicated state-of-the-art architectures in terms of average test accuracy (Acc), individual test accuracy for each environment, and the average time required to identify the environment (prediction time).

Table 7. Comparison between our model and state-of-the-art alternatives.

Architecture	H-NLOS Acc (%)	H-LOS Acc (%)	R-LOS Acc (%)	U-LOS Acc (%)	U-NLOS Acc (%)	Acc (%)	Prediction Time (µs)
Our Model	99.9	95.2	92.7	97.4	97.2	96.48	39.56
ResNet50	98.1	88.2	77.8	90.1	93.5	89.54	672
Xception	97.8	91.7	81.4	91.2	94.5	91.32	794
InceptionV3	99.1	79.8	86.9	96.1	93.9	91.08	683
Inception ResNetV2	98.5	89.1	80	86.5	95.8	89.98	1621
DenseNet201	98.5	92.7	85.7	91.2	96.6	92.94	1349
MobileNetV2	96.8	77.8	96	58.2	65.5	78.86	318
DCNN [37]	98.9	96.9	94.3	95.8	99.2	97.02	125

From Table 7, it is clear that our proposed model has the best performance in terms of prediction time compared to all the other architectures presented in the table, with a prediction time of 39.56 µs. The architectures ResNet50, Xception, and InceptionV3 have prediction times around of 700 µs, whereas the InceptionResNetV2, DenseNet201, MobileNetV2, and DCNN [37] architectures have 1621 µs, 1349 µs, 318 µs and 125 µs, respectively. Thus, the prediction time achieved by our model is significantly (at least three-fold) lower than the prediction time attained by the other architectures.

Regarding the overall test accuracy (Figure 12), our model reaches 96.48%. This achieved accuracy is greater than the test accuracy attained by ResNet50, Xception, InceptionV3, InceptionResNetV2, DenseNet201, and MobileNetV2, which have 89.54%, 91.32%, 91.08%, 89.98%, 92.94%, and 78.86%, respectively. Although the proposed DCNN architecture in [37] scores slightly higher than our model in term of test accuracy (by 0.54%), the prediction time achieved by our model is three times faster than DCNN [37].

To attain more insight into the classification performance of the considered architectures, the individual test accuracy of each vehicular environment is presented in Table 7. It can be seen that all the architectures are able to successfully discriminate the H-NLOS environment with accuracy of more than 96%; our model has the best accuracy, at 99.9%. Concerning the H-LOS environment, the individual accuracies are acceptable (close to 80%) for MobileNetV2 and InceptionV3, and the other architectures provide good individual accuracies of around 90%. The test accuracies for the R-LOS environment are less significant compared to other environments. This is due to the channel characteristics of this environment, which are close to the channel characteristics of the U-LOS and H-LOS environments. However, the accuracy for the R-LOS environment is considered good for our model, MobileNetV2, and DCNN (up to 90%), and is acceptable for the other models (around 80%). For the U-LOS environment, the test accuracy is generally around 90%, while our proposed model has the best accuracy at 97.4%. For the U-NLOS environment, all the architectures except MobileNetV2 are able to recognize this environment with high accuracy (up to 93%).

5.4. Minimum Performance Overhead and Reliability

Because our proposed vehicular environment identification approach based on CSI is intended to be implemented in autonomous connected cars for use in a time-critical setting, it is important that the environment identification prediction time consistently meet latency requirements in every scenario. According to [38], this hard time limit typically falls within a few milliseconds. As seen in the results, our proposed CSI-based vehicular environment identification model demonstrates a prediction time that is consistently within the **microsecond** range, which is well under the required range time.

In addition, our proposed CNN architecture using the CSI values as an inputs features proves that it can reliably identify different vehicular environments with high accuracy (96, 48%) that meets the requirements for autonomous connected cars.

6. Conclusions

In this paper, we have presented a deep learning-based vehicular environment identification approach using vehicular wireless channel characteristics in the form of estimated CSI as input features for a CNN model. The results of our validation tests have demonstrate that our methodology can reliably recognize the surrounding environment with high accuracy (96.48%). These same results show that our approach has minimal performance overhead, measured in microseconds, which is well within the expected operational range across various autonomous driving scenarios. In addition, our CNN model has comparable performances to existing state-of-the-art architectures. In summary, our CSI-based vehicular environment identification approach is validated as a reliable solution to enable speed limit decisions for autonomous vehicles. However, because our vehicular environment identification method is based on the use of channel characteristics, it requires a continual exchange of messages. Moreover, we demonstrate promising results with our simulation scenarios. Hence, real-world CSI-based vehicular environment identification testbeds on a larger scale involving more vehicles in specific road conditions remains an open research problem worth investigating in the future.

Author Contributions: Conceptualization: S. R., R. S., Y. E., A. R. and A. H.; methodology: S. R., R. S. and A. H.; software: S. R.; validation: A. H.; formal analysis: S. R. and R. S.; data curation: S. R. and R. S.; writing—original draft preparation: S. R and R. S.; writing—review and editing: S. R. and A. H.; supervision: A. H.; funding acquisition: A. R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement:

Informed Consent Statement:

Data Availability Statement:

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

References

1. Kockelman, K.; Boyles, S.; Stone, P.; Fagnant, D.; Patel, R.; Levin, M.W.; Sharon, G.; Simoni, M.; Albert, M.; Fritz, H.; et al. *An Assessment of Autonomous Vehicles: Traffic Impacts and Infrastructure Needs*; Technical Report; University of Texas at Austin, Center for Transportation Research: Austin, TX, USA, 2017.
2. Uhlemann, E. Time for autonomous vehicles to connect [connected vehicles]. *IEEE Veh. Technol. Mag.* **2018**, *13*, 10–13.
3. Yue, L.; Abdel-Aty, M.; Wu, Y.; Wang, L. Assessment of the safety benefits of vehicles' advanced driver assistance, connectivity and low level automation systems. *Accid. Anal. Prev.* **2018**, *117*, 55–64.
4. Nahar, A.; Das, D. MetaLearn: Optimizing routing heuristics with a hybrid meta-learning approach in vehicular ad-hoc networks. *Ad Hoc Netw.* **2023**, *138*, 102996.
5. Abuelsamid, S. Toyota has big plans to get cars talking to each other and infrastructure in the US. *Forbes, April.* **16** pp. 2018 (2018)
6. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. Deepdriving: Learning affordance for direct perception in autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2722–2730.
7. Okamoto, K.; Itti, L.; Tsiotras, P. Vision-based autonomous path following using a human driver control model with reliable input-feature value estimation. *IEEE Trans. Intell. Veh.* **2019**, *4*, 497–506.
8. Zhang, Y.; Sun, P.; Yin, Y.; Lin, L.; Wang, X. Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1251–1256.
9. Seo, Y.W.; Lee, J.; Zhang, W.; Wettergreen, D. Recognition of highway workzones for reliable autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 708–718.
10. Sauer, A.; Savinov, N.; Geiger, A. Conditional affordance learning for driving in urban environments. *arXiv* **2018**, arXiv:1806.06498.
11. Kim, B.; Kim, D.; Park, S.; Jung, Y.; Yi, K. Automated complex urban driving based on enhanced environment representation with GPS/map, radar, lidar and vision. *IFAC-PapersOnLine* **2016**, *49*, 190–195.
12. Varga, R.; Costea, A.; Florea, H.; Giosan, I.; Nedevschi, S. Super-sensor for 360-degree environment perception: Point cloud segmentation using image features. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–8.
13. Duan, X.; Jiang, H.; Tian, D.; Zou, T.; Zhou, J.; Cao, Y. V2I based environment perception for autonomous vehicles at intersections. *China Commun.* **2021**, *18*, 1–12.
14. Lee, D.H.; Chen, K.L.; Liou, K.H.; Liu, C.L.; Liu, J.L. Deep learning and control algorithms of direct perception for autonomous driving. *Appl. Intell.* **2021**, *51*, 237–247.
15. Florea, H.; Petrovai, A.; Giosan, I.; Oniga, F.; Varga, R.; Nedevschi, S. Enhanced perception for autonomous driving using semantic and geometric data fusion. *Sensors* **2022**, *22*, 5061.
16. Kabir, M.F.; Roy, S. Real-time vehicular accident prevention system using deep learning architecture. *Expert Syst. Appl.* **2022**, *206*, 117837.
17. Zhu, H.; Yuen, K.V.; Mihaylova, L.; Leung, H. Overview of environment perception for intelligent vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2584–2601.
18. Ribouh, S.; Phan, K.; Malawade, A.V.; El Hillali, Y.; Rivenq, A.; Al Faruque, M.A. Channel State Information Based Cryptographic KeyGeneration for Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* 2020. *to appear*.
19. Alexander, P.; Haley, D.; Grant, A. Cooperative intelligent transport systems: 5.9-GHz field trials. *Proc. IEEE* **2011**, *99*, 1213–1235.
20. Wan, J.; Lopez, A.B.; Al Faruque, M.A. Exploiting wireless channel randomness to generate keys for automotive cyber-physical system security. In Proceedings of the 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS), Vienna, Austria, 11–14 April 2016; pp. 1–10.
21. ETSI, T. 103 257-1 V1. 1.1 (2019-05) Intelligent Transport Systems (ITS). Access Layer.
22. Bernado, L.; Zemen, T.; Tufvesson, F.; Molisch, A.F.; Mecklenbräuker, C.F. Delay and Doppler spreads of nonstationary vehicular channels for safety-relevant scenarios. *IEEE Trans. Veh. Technol.* **2013**, *63*, 82–93.

23. Tan, I.; Tang, W.; Laberteaux, K.; Bahai, A. Measurement and analysis of wireless channel impairments in DSRC vehicular communications. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 4882–4888.
24. Zemen, T.; Bernadó, L.; Czink, N.; Molisch, A.F. Iterative time-variant channel estimation for 802.11 p using generalized discrete prolate spheroidal sequences. *IEEE Trans. Veh. Technol.* **2012**, *61*, 1222–1233.
25. Zhuang, Y.; Hua, J.; Wen, H.; Meng, L. An iterative Doppler shift estimation in vehicular communication systems. *Procedia Eng.* **2012**, *29*, 4129–4134.
26. Ghanavati, A.Z.; Pareek, U.; Muhaidat, S.; Lee, D. On the performance of imperfect channel estimation for vehicular ad-hoc networks. In Proceedings of the 2010 IEEE 72nd Vehicular Technology Conference-Fall, Ottawa, ON, Canada, 6–9 September 2010; pp. 1–5.
27. Sutar, M.B.; Patil, V.S. LS and MMSE estimation with different fading channels for OFDM system. In Proceedings of the 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; Volume 1, pp. 740–745.
28. Kukačka, J.; Golkov, V.; Cremers, D. Regularization for deep learning: A taxonomy. *arXiv* **2017**, arXiv:1710.10686.
29. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. 2016. *arXiv* **2016**, arXiv:1603.05027.
32. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
33. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
34. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261.
35. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
37. Elwekeil, M.; Wang, T.; Zhang, S. Deep Learning for Environment Identification in Vehicular Networks. *IEEE Wirel. Commun. Lett.* **2019**, *9*, 576–580.
38. Dixit, V.V.; Chand, S.; Nair, D.J. Autonomous vehicles: Disengagements, accidents and reaction times. *PLoS ONE* **2016**, *11*, e0168054.