Applications of Deep Learning in Intelligent Transportation Systems

Arya Ketabchi Haghighat, Varsha Ravichandra-Mouli, Pranamesh Chakraborty, Yasaman Esfandiari, Saeed Arabi, Anuj Sharma

Abstract

In recent years, Intelligent Transportation Systems (ITS) have seen more efficient and faster development through the implementation of Deep Learning (DL) techniques, especially in problem domains previously addressed using analytical or statistical solutions. Improvements resulting from DL applications have facilitated better traffic management and traffic planning, increased safety and security on transit roads, decreased maintenance costs, and more optimized public transportation and ride-sharing company performance, in addition to having advanced driverless vehicle development to a new level. This paper's primary objective is to provide a comprehensive review and insight into the applications of DL models on ITS, accompanied by presentation of the progress in ITS research that has been due to DL studies. First, different DL techniques are discussed, and then an in-depth analysis and explanation of the current applications of these techniques on transportation systems is provided. This paper's outline of the impact of DL on ITS demonstrates the significance of DL in the ITS domain. Finally, different embedded systems used for deployment are described and their advantages and disadvantages are discussed. Through this paper's systematic review, the credible benefits of DL models on ITS are delineated and key directions for future DL research in relation to ITS are identified.

Keywords: Deep Learning, ITS, survey, Transportation systems

1. Introduction

The emergence of machine learning and its substitution for several statistical models have led to better problem-solving, which in turn has led various fields of study to turn their research paths to take advantage of this new method. Transportation systems have been influenced by the growth of machine learning, particularly in Intelligent Transportation Systems (ITS). With the proliferation of data and advancements in computational techniques such as Graphical Processing Units (GPUs), a specific class of machine learning known as Deep Learning (DL) has gained popularity. The capability of DL models to address large amounts of data and extract knowledge from complex systems has made them a powerful and viable solution in the domain of ITS. A variety of networks in DL have helped researchers to formulate their problems in a way that can be solved with one of these neural network techniques. Traffic signal control for better traffic management, increasing the security of transportation via surveillance sensors, traffic rerouting systems, health monitoring of transportation infrastructure and several other problems now have a strong new approach, and for several challenging problems in transportation engineering, new solutions have been created.

There have been several surveys of the literature on the application and enhancement of ITS using DL techniques. However, most of these have tended to focus on a specific aspect of DL or a specific aspect of ITS. For instance, [1] conducted a survey of big data analytics in ITS. A review of computer vision playing a key role in roadway transportation systems was discussed in [2]. While [3] reviews DL models across the transportation domain, it is not a comprehensive survey that encompasses all current research publications on the ITS domain and DL. One dedicated review on enhancing transportation systems via DL was done in [4] where substantial research was included, but it focused primarily on traffic state prediction and traffic sign recognition tasks. The ITS domain includes other tasks, such as public transportation, ride-sharing, vehicle re-identification and traffic incident prediction and inference tasks, which are all represented in this paper to make its extent more comprehensive and holistic. The transportation and research community has always taken notice of pivotal research directions, with the earliest review of neural nets applied to transportation [5], where the critical review spanned the classes of problems, neural nets applied and the challenges in addressing various problems. It is this that motivates of the question we address in this paper: *How effective and efficient are the current DL research applications for the domain of ITS?* To the best of the authors' knowledge, the literature in this field has suffered from the lack of a holistic survey that takes a broader perspective of ITS as a whole and its enhancement using DL models.

The purpose of this paper is therefore to present the systematic review we have conducted on the existing state of the research on ITS and its foray into DL. In section 2, we discuss our approach taken to identify relevant literature. In section 3, we talk about different methods of DL network systems and breakthrough research on those methods. In section 4, we talk about different applications of DL methods in transportation engineering, specifically 6 major application categories in ITS.

In section 5, we investigate different available embedded systems, or devices that can facilitate the running of neural network experiments. Finally, in section 6, we provide a summary and an outlook for future research.

The research methodology which is followed in this paper is PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses)[6]. Following this method, we first produced a questionnaire and in each paper we reviewed, we looked for answers to these questions. The focus of these questions is about the gap which each paper tries to address, their proposed solutions and finally the performance of these solutions for their datasets.

2. Research Approach and Methodology

This paper performs a detailed analysis of existing studies on Intelligent Transportation Systems (ITS) and Deep Learning (DL). Articles were searched in multiple databases using the search strategy described below. The collected articles were then reviewed and organized. The scope of this review was restricted to conference proceedings and journal articles, including existing literature reviews.

Relevant articles were primarily obtained by querying the TRID TRB database [7], where the search terms included "deep learning", "convolutional". These search terms were sought in the title, abstract and notes. Then the references of the papers identified were examined to trace other trusted journals and papers. Also, online searches on various databases such as Scopus, ScienceDirect, IEEE, and ArXiv were done. All papers obtained were included in this review if they met the following criteria:

- Describe solutions to ITS problems using DL, as identified by methodology sections that include DL-based model development
- Published between January 2015– October 2019 (during which period the majority of research so far using DL in ITS has been conducted)
- Not a book, book chapter, dissertation, thesis or technical report
- Not a general introduction to ITS
- Not in the domain of autonomous vehicles

Though DL boom was spawned by the ImageNet project in 2012 [8] and applications of DL on ITS first appeared in 2013, substantial growth in ITS research by means of DL methodologies didn't started until 2015. This is illustrated in Figure 1. Since then, there has been a steady growth in the prominence of DL-based ITS studies across journals and conferences. In the year 2019, up until October, 43 papers have been published across various ITS applications. In light of the marked increasing importance of DL as an ITS research method, in the following section, we will discuss and review the various DL structures and then their key applications in the ITS domain.

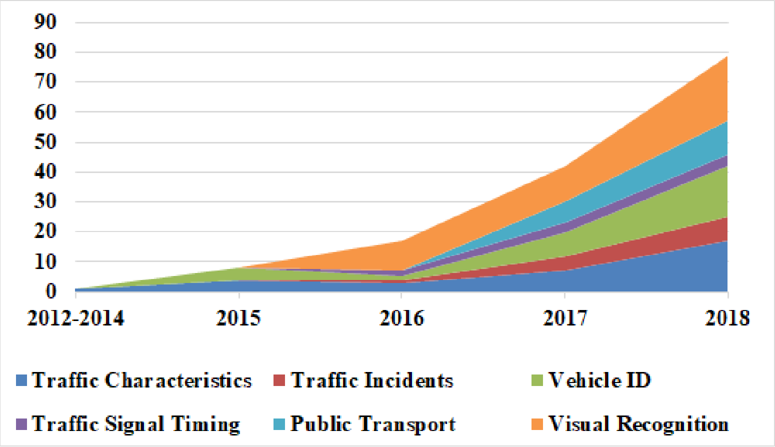


Figure 1: Year-wise publication growth in ITS domains

3. Background on techniques in deep learning

3.1. Deep Neural Networks (DNN)

Deep learning (DL) is a specific subcategory of machine learning where several layers of stacked parameters are used for the learning process [9]. These parameters are component representations of different aspects which can affect the result of the network. Each layer contains several perceptrons (known also as neurons or hidden units) which carry weights for the parameter. The input of each layer is multiplied by these parameters and therefore the output is a representation of the impact of each parameter on the input. Usually after each layer or several layers of neurons, a nonlinearity function such as the tanh, sigmoid, or rectified linear function (ReLU) [10] is used to generate the output layer. All these layers combine to form a Deep Neural Network (DNN) [11]. There are two major challenges in building a DNN: first, designing the structure of the network, which includes the number of layers, number of neurons in each layer, and nonlinearity function type; and second, adjusting the weight of the parameters to train the network on how it should perceive the input data and calculate the output. For the first challenge, what is usually most helpful is simply trial and error and overall experience. For the second challenge, the back-propagation method is the most popular method to train the weight of parameters in a supervised manner. More detail about this method can be found in [11]. Although all the techniques which will be discussed in the rest of this paper can be classified as a subcategory of DNN, here in this paper, DNN is defined as the most simple structure of a network, in other words, fully connected layers. In this fully connected model, there is a connection between all the neurons of one layer to all the neurons in another layer, and for each connection, there is a weight which should be determined through back-propagation method.

3.2. Convolutional Neural Networks (CNN)

One of the major applications of neural networks was Computer-Aided Detection (CAD) that aimed to increase classification accuracy and inferencing time. A revolutionary method was proposed in [12] called Convolutional Neural Networks (CNN). Inspired by the vision system of cats which are locally sensitive and orientation-selective, as presented in [13], [12] suggested that instead of using fully connected layers of neural networks, it is possible to use a single kernel with shared weights to wisp the entire image and extract the local features. The proposed method enhanced the detection effectiveness both in terms of accuracy and memory requirement when compared with traditional methods, which required handcrafted feature extractions [14].

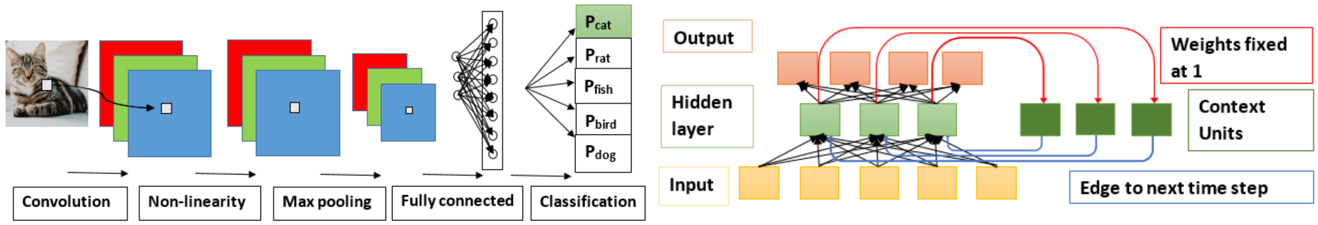
CNN is a detection architecture that automatically learns spatial hierarchical features using back-propagation through the network. A schematic figure of this architecture is presented in Figure 2a. These networks usually contain three types of layers: convolution, pooling, and fully connected, where the first two are used to extract the features and the last one used as a classifier [15].

The convolution layer consists of a combination of a convolution kernel, which counts as a linear part of the layer, and a nonlinear activation function. The main advantage of using a kernel that shares weights in operation, is extracting the local features and learning the spatial hierarchies of features efficiently by reducing the required parameters. Then the nonlinear activation function maps the results onto the feature map. In order to reduce the number of parameters, usually one pooling layer comes after a few convolutional layers in order to downsample the data, by taking the maximum unit (max pooling) or the average (average pooling) of a collection of units and substituting it as a representative of these collections. After extracting features and downsampling the data by the convolution and pooling layers, they are mapped onto the final output by fully connected layers. The output of these layers usually is the same size as the number of classes and each output indicates the probability of it belonging to that class. Finally, this string maps onto the final result by an activation function. This activation function can be sigmoid for binary/multiclass classification; softmax for single/multiclass classification or to identity continuous values in case the of regression [16].

Based on the fact that in order to train a deep model a large amount of data is needed, CNN and other models' popularity only began to rise when a large quantity of labeled data were provided for the ImageNet challenge [8]. Afterward, lots of architectures have been proposed which use these CNN blocks to enhance the efficiency of CAD. Some of these methods are AlexNet, Inception, VGGNet 16/19, Resnet, and etc. However, in order to increase the accuracy of detection, other concepts have been used in the process. Some of these concepts are transfer learning, which uses the knowledge of the network from pretraining on a large dataset in order to train the network on a smaller dataset [16]. The other method is training with an equal prior instead of a biased prior in those cases where the dataset has a bias towards one of the classes (imbalanced dataset). In this case, different sampling or resampling rates are applied to the dataset to balance it. The effect of these different methods of changing the architecture, using transfer learning and balancing the dataset for various datasets are investigated in [17].

3.3. Recurrent Neural Networks (RNN)

Recurrent neural networks (RNNs), another class of supervised DL models, are typically used to capture dynamic sequences of data. RNNs can successfully store the representation of recent inputs and capture the data sequence by introducing



(a) CNN Schematic

(b) RNN Schematic

Figure 2: Figures depicting CNN and RNN schematic

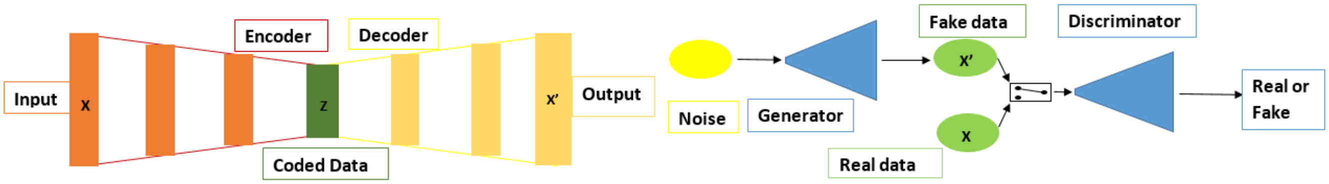
a feedback connection to interpret the data. This ability can play the role of memory to pass information selectively across sequence steps to process data at a certain time. Thus, each state depends on both the current input and the state of the network at a previous time. In other words, there is a similarity between a traditional, simple RNN and Markov models [18]. In 1982, the first algorithm for recurrent networks was used by [19] in order to do pattern recognition. In 1990, [20] introduced his architecture, which is known as the most basic RNN. A schematic figure of this architecture is presented in Figure 2b. In this architecture, associated with each hidden unit, there is a context unit which takes the exact state of the corresponding unit at the previous time as an input and re-feeds it with the learned weight to the same unit in the next step.

Although training RNN networks seems to be straightforward, vanishing or exploding gradient problems remain the two main difficulties. These problems can happen during learning from previous states when the chain of dependencies gets prolonged and, in this case, it is difficult to choose which information should be learned from past states. In order to solve the problem of an exploding gradient in recurrent networks, which can result in oscillating weights, [21] has suggested Truncated Back-Propagation Through Time (TBPTT), which sets a certain number of time steps as a propagation limit. In this case, to prevent exploding the gradient, a small portion of previously analyzed data is collected to use during the training phase. However, this means that in the case of long-range dependencies cases, the former information related to these dependencies will end up lost.

Long Short Term Memory (LSTM) architecture has been suggested by [22] to solve both these problems together. The primary idea of this method is using a memory cell with only two gates of input and output. The input gate decides when to keep the information in the cell and the output gate decides when to access the memory cell or prevent its effect on other units. In recent years, several corrections and improvements have been made on LSTM architecture.

As described above, LSTM contains a memory cell that holds its state over time, and based on its regulation, controls how this cell affects the network. The most common type of LSTM cell has been suggested by [23]. Several gates and components which are added to this cell are different from the basic suggested LSTM by [22]. A logistic sigmoid function is usually used as the gate activation, though due to the state-of-the-art design of [23], a tanh function is usually used as the block input activation and block output activation. The forget gate and peephole connections were first suggested by [24] that enables the cell to reset by forgetting its current state and passing the current state data from the internal state to all gates without passing them through an activation function.

Finally, it is notable that [25] has proposed a gated recurrent unit (GRU) inspired by the LSTM block, where they have eliminated the peephole connections and output activation function. They have also coupled the input gate and forget gate into one gate called the update gate and what passes through their output gate is only recurrent connections to the block input. This architecture is much simpler than LSTM and based on what it eliminates, it avoids a significant reduction in performance, which makes it more popular to use.



(a) AE Schematic

(b) GAN Schematic

Figure 3: Figures depicting AE and GAN schematic

3.4. Autoencoders (AE)

One of the most important task in DL is access to a large amount of data to train the model. Usually, such a dataset is not readily available and producing a rich dataset would be expensive. In this situation, unsupervised methods show their value. Instead of training models using labeled data, unsupervised methods extract the features of unlabeled data and use these extracted features to train the model. Autoencoders (AEs) are one such method which aims to reconstruct the input data and in this manner is similar to principal component analysis. AEs are composed of two networks that are concatenated to each other. The first network extracts and encodes the input data into its main features and the second network use these features to reshape arbitrary random data to reconstruct something similar to the input data. The schematic figure of this architecture is presented in Figure 3a. Although the concept of AEs has been used previously as a denoiser [26] and data constructor [27], it found a new application as variational AEs [28]. To minimize the difference from input and output, [28] have used the variational inference method. They introduced a lower bound on the marginal likelihood and tried to maximize it to minimize the error between input and output. [29, 30] have explained exactly how a variational AE can be built.

Usually, an AE's hidden layer is smaller than its input layer, although the opposite situation can happen as well. Also, the horizontal orientation of AEs are defined as combining two or more AEs horizontally, and this can have different motivations such as different learning algorithms (e.g., RBM, neural network, or Boolean) or different initialization and learning rates. In addition to details about these situations, linear and nonlinear AEs have been studied by [31]. It has been shown that a Boolean AE as a nonlinear type has the ability to cluster data and an AE layer on top can be used as a pretrainer for a supervised regression or classification task.

3.5. Deep Reinforcement Learning (DRL)

Reinforcement Learning (RL) attempts to train a machine to act as an agent who can interact with the environment and learn to optimize these interactions by learning from responses [32]. In RL, the agent observes the environment and gets a state signal and chooses an action that impacts the environment to produce a new state. In the next step, a reward from the environment and the new state is fed to the agent to help it decide more intelligently in the next step. The goal of an agent in this setup is gaining the maximum reward over the long term by following an optimal policy. The algorithm of RL is usually based on the Markov Decision Process (MDP) [33]. The problems that can be solved by RL algorithms can be divided into episodic and non-episodic MDP. In episodic MDP, the state will reset at the end of the episode and the return (accumulation of rewards for the episode) is calculated. In non-episodic MDP, there is no end of the episode and using a discount factor is vital to prevent an explosion of return values [32].

There are two functions usually used in RL: the state-value function, also known as the value function, is the expected return if the agent starts at a given state (no action limitation), whereas the action-value function, also known as the quality function (Q-function) is the expected return of starting at a given state and taking a particular action. Usually, one of two methods is implemented to solve an RL problem. In the first approach, the Q-function is predicted using different methods of temporal difference controls such as state–action–reward–state–action (SARSA), which improves the estimation of Q. The second approach is Q learning, which directly approximates the optimal Q. Both of these methods use bootstrapping and learn from incomplete episodes.

Deep Reinforcement Learning (DRL) is an approach to solving the RL problem using a DNN. Although the history of DRL began in the 1990s when [34] developed a neural network that reached an expert level in backgammon, its rebirth can be considered as [35] who introduced Deep Q-Networks (DQN) as DNNs that can approximate Q instead of reading its value from a Q table that indicates for each state what the Q value would be for taking each action. In this new method, complex and high dimensional problems have potential to be addressed easily [35]. The model used by [35] extracted images from the Atari games and used a combination of a CNN model and a fully connected layer on the data extracted from the images to obtain an estimate of the Q value.

However, because of the complexity of DRL, it can be unstable. Therefore, much research has been focused on solutions able to defeat this instability. Experience replay [36] and target networks [35] are the two most used techniques to make RL stable. Other techniques include Double-Q learning [37] and dueling DQN [38], which have also been proposed to make DRL more robust and stable. In Double-Q learning, the second estimator is used for estimating an extra assumptive Q' to approximate the Q value more precisely. On the other hand, dueling DQN [38] uses a baseline instead of an accurate calculation of Q value to learn relatives.

3.6. Generative Adversarial Networks (GAN)

Generative adversarial networks (GANs) are a specific class of deep learning networks that learn how to extract the statistical distribution of training data to synthesize new data similar to real-world data. These synthetic data can be used for several applications such as producing high-resolution images [39], denoising low-quality images, and image-to-image translation [40]. Most of the generative models use the maximum likelihood concept to create a model that can estimate the probability distribution of the training data and synthesize a dataset that maximizes the likelihood of the training data.

Although calculating maximum likelihood can directly result in the best action of the model, sometimes these calculations are so difficult that it is more beneficial to implicitly estimate this amount. In the case of explicit density calculation, three main types of models are popular:

- Fully Visible Belief Networks
- Variational AEs
- Markov Chain Approximations

All of these models, however, suffer from the problems of low speed, low quality, and early stoppage [41]. To overcome these problems, [42] has suggested a method that does not require explicit definition of the density function. This model can generate samples in parallel; no Markov chain is needed to train the model and no variational bound is needed to make it asymptotically consistent.

This method has two models: the generative model which is responsible to pass random noise through a multilayer network to synthesize samples, and the discriminative model, which is responsible to pass real data and artificial data through a multilayer network to detect whether the input is fake or real. A schematic figure of this architecture is presented in Figure 3b. Both models use back-propagation and dropout algorithms: the generative model to create more realistic data and the discriminative model to achieve better distinction between real and fake data.

When GANs were first proposed in both their generative and discriminative models, fully connected networks were used. However, later in 2015, [43] suggested a new architecture named deep convolution GAN (DCGAN), which uses batch normalization in all layers of both models, except the last layer of the generator and first layer of the discriminator. Also, no pooling or unpooling layer is used in this architecture. A DCGAN allows the model to understand operations in latent space meaningfully and respond to these operations by acting on the semantic attributes of the input [41].

The other improvisation on the GAN architecture has been conditional GAN [44], where both networks are class-conditional, which means the generator tries to generate image samples for a specific class and the discriminator network is trained to distinguish real data from fake data, conditional on the particular class. The advantage of this architecture is better performance in multimodal data generation [45].

In the next section, we discuss and review the applications of deep learning models to transportation.

4. Applications in Transportation

4.1. Performance Evaluation

Before reviewing papers that have already used DL methods to investigate ITS applications, it is necessary to make clear the model evaluation criterias used. The classification metrics are Accuracy (AC), Precision (PR), Recall (RL), *Top 1* accuracy and *Top 5* accuracy, while the regression metrics are Mean Average Precision (mAP), Mean Absolute Error (MAE), Mean Absolute Percentage Error ($MAPE$) and Root Mean Squared Error ($RMSE$).

$$AC = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$PR = \frac{TP}{TP + FP} \tag{2}$$

$$RL = \frac{TP}{TP + FN} \tag{3}$$

where TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative.

Top 1 accuracy means the model's top answer must match the expected answer.

Top 5 is when at least one of the model's five highest probability answers must match the expected answer.

mAP is the mean of the average precision (AP) scores for every query, where AP is the area under the PR vs RL curve.

IoU is the ratio between area of overlap and area of union, between the predicted and the ground truth bounding boxes.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_i - \overline{y_i}| \tag{4}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|\frac{y_i - \overline{y_i}}{y_i}| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_i - \overline{y_i})^2} \tag{6}$$

$$MSRE = \frac{1}{n}\sum_{t=1}^{n}(\frac{y_i - \overline{y_i}}{y_i})^2 \tag{7}$$

where y_i is the actual value of observed travel time, $\overline{y_i}$ is the predicted value of travel time, and n is the number of observations.

We now discuss different applications of deep learning in ITS. The included topics have been selected based on the functional areas in ITS as mentioned in [46] and have been studied substantially over the period of 2012-2019.

4.2. *Traffic characteristics prediction*

One of the most considered applications of DL in transportation is related to traffic characteristics prediction. Traffic characteristics information can help drivers to choose their routes more wisely and traffic management agencies to manage traffic more efficiently. The main characteristics of interest are traffic flow, traffic speed and travel time. Since these characteristics are not mutually exclusive, methods that are used to predict one of them also can be used to predict the value for the remaining features. Due to this, methods used to make these predictions are discussed together as follows.

Based on the duration of prediction for each traffic characteristic, a forecast value is usually classified as short-term (S) for predictions within less than 30 minutes, medium-term (M) for a prediction window between 30 to 60 minutes, and long-term (L) within more than 60 minutes [47]. Since driving behavior and traffic characteristics can vary across locations, results from one dataset are difficult to apply to other datasets [4]. Previously, traffic feature prediction has predominantly used parametric and statistical methods, such as autoregressive integrated moving average (ARIMA) modeling, but most of the time these methods have been incapable of predicting irregular traffic flows [4]. However, through the emergence of machine learning and furthermore DL methods, nonparametric methods are now being used in traffic characteristics prediction to achieve higher accuracy.

One of the first attempts to predict traffic characteristics has used deep belief networks (DBN) as an unsupervised feature learner. [48, 49, 50] have implemented DBNs for traffic flow prediction. [51, 52] have used the same concept for predicting travel time and traffic speed. Along with traffic data, weather data has been fed into DBNs using data fusion techniques to predict traffic flow more accurately [53].

However, due to the nature of the above mentioned traffic features and their dependency on past traffic conditions, several studies have been done to discover correlations using RNN to predict traffic characteristics. For instance, [54] have used a gated RNN unit to predict traffic flow with respect to the weather conditions, where [55] have used LSTM to overcome the same challenge. [56, 57] have used LSTM to predict travel time as well as traffic flow, while also taking into account weather conditions. Finally, [58] have implemented a combination of deep RBM and RNN to predict congestion in transportation network links.

[59] have tried to increase the AC of traffic flow prediction especially for nonrecurrent traffic congestion, such as a special event or harsh weather, by paying more attention to the spatiotemporal feature of traffic. This feature is grounded in the assumption that to predict any traffic characteristic, we need both the historical data on that particular location and current traffic in the neighboring areas. To accomplish this, [60] have tried to combine an RNN with a CNN to pay attention to both the temporal and spatial aspects of traffic. [61, 62, 63] have combined the power of LSTM + CNN to understand both temporal and local dependencies to predict different traffic characteristics. [64] have considered two challenges, the first being the dynamic dependency of traffic on temporal features, that is, in different hours of the day, this dependency may differ from one direction of traffic flow to another direction. The second challenge has been the probability of shifting time periods in relation to traffic density. In other words, a periodic temporal dependency may shift from one time to another (e.g., on different days of the week). As a result [64] designed a network consisting of a flow-gated local CNN network to capture the dynamic of the spatial dependencies and an LSTM network as a periodically shifted attention mechanism for handling the periodic dependencies. One other approach to accounting for both types of dependencies was taken by [65]. They converted data into images representing the two dimensions of time and space. By converting their data matrices into images, they were able to use a CNN model to extract image features and predict the network-wide traffic speed. [66] improved this approach later by adding a temporal gated convolution layer to extract temporal features.

To extract both spatial and temporal features, [67] have used a deep model called the stacked bidirectional and unidirectional LSTM (SBU-LSTM) model where the bidirectional LSTM considers both the backward and forward dependencies in time-series data. Since traffic conditions have periodicity, by analyzing both backward and forward features, the AC can be increased.

One of the other models able to consider the spatiotemporal property of traffic has been AE, which was proposed first by [68] and improved by [69] using denoising Stacked AE (dSAE) and [47] by combining LSTM and AE to predict traffic conditions at peak hours and in post-accident situations. To predict post-accident situations, they extracted a latent representation of the static features that are common in all accidents from stacks of AE and combined this with a temporal correlation to traffic flow that came from stacks of LSTM, using a linear regression (LR) layer.

Table 1 summarizes all these papers, with the columns from left to right describing for each study the traffic characteristics investigated and its DL model, dataset, experiment results (best results achieved), baseline model and the baseline model's best results, prediction window length, hyperlink to the given paper and its year of publication.

To the best of the authors' knowledge, all studies matching the meta-analysis criteria described in section 2 of the current paper related to travel time, traffic speed, traffic flow, traffic conditions, and traffic density have been tabulated here. For traffic conditions, the goal was to predict if the road is congested or not. Results performed on multiple datasets are also represented in Table 1. To have uniformity, the best results are those achieved when the window length is 'S' (Short-term). This table structure is followed across all tables in this paper.

Table 1: Overview of papers using deep learning techniques for traffic characteristic prediction.

Characteristic	Model	Dataset	Experiment Results		Baseline Model Results			Length	Paper	Year
			MAPE%	Others	MAPE%	Others	Model			
Travel time	AE	Simulated data	5.79					M	[70]	2015
	DBN	PeMS	3.00					S,M,L	[51]	2017
	LSTM+DNN	PeMS	0.961		1.006		Ridge Reg.	S	[56]	2018
	LSTM	St.Louis transport net	7.09		9.8		Instantanious travel time	L	[52]	2018
Traffic speed	DBN	Beijing Arterial	5.809		5.968		BP-NN	S,M	[55]	2016
	eRCNN	Beijing Ring	0.19		0.21		CNN	S	[60]	2017
	CNN	Beijing Ring		AC:93.21%		AC:91.7%	OLS	S	[65]	2017
	SBU+LSTM	INRIX	5.674		6.3		Random Forest(RF)	S	[67]	2018
	DLSTM	G42 toll data	10.8					S	[71]	2019
	Bi-LSTM	AVI,Xuancheng		MSE:7.32		MSE:10.51	LSTM NN	S,M	[72]	2019
	CNN+LSTM	Rozelle road segment		RMSE:4.46(km/h)		RMSE:4.48	LSTM	S,M	[73]	2019
	CNN	Traffic Centre, Seoul		RMSE:2.48(km/h)		RMSE:2.75	LSTM	S,M,L	[74]	2019
	LSTM+GRU	RTMS, Beijing	5.85		8.61		LSTM	L	[75]	2019
	GCN	Beijing Ring	9.57		10.25		LSTM	S	[76]	2019
Traffic flow	DBN	PeMS EESH		AC:90%		AC:86%	NN	S,M,L	[49]	2014
	SAE	PeMS	6.75		7.4		RBF NN	S,M,L	[68]	2015
	LSTM	PeMS	6.49		7.63		SAE	S,M,L	[57]	2015
	DBN	PeMS		RMSE:0.06(V/15min)		RMSE:0.063	ANN	L	[53]	2016
	DSAE	PeMS	23.2		23.7		NN	L	[69]	2016
	LSTM+DBN	Beijing Ring	11.69		12.84		LSTM	S,M	[77]	2017
	LSTM+SAE	PeMS	5		10		Random Walk	S,M,L	[47]	2017
	CNN	PeMS		RMSE:0.0061				L	[61]	2017
	LSTM	Beijing Arterial	6.05		6.32		SAE	S,M,L	[78]	2017
	RNN+CNN	PeMS		RMSE:0.028(V/5min)		RMSE:0.030	LSTM	L	[62]	2018
	DBN	Visum Simulator	8.75		9.39		ARIMA-PSO	S,L	[63]	2018
	ConvLSTM	Beijing Traffic		RMSE:6.95(km/h)		RMSE:7.359	LSTM	L	[79]	2018
	DNN	PeMS		RMSE:27.91(V/5min)		RMSE:29.84	DeepST	M	[80]	2018
	CNN	Washington Interstate		RMSE:5.5(V/5min)		RMSE:15	DSAE	L	[81]	2018
	DBN	PeMS	3.19		5.32		SAE	S,M,L	[82]	2018
	CNN	BJER4+PeMS	9.11		9.31		Graph GRU	S,M	[83]	2018
	LSTM+CNN	NY Bike/Car	16.3		17.36		DMVST-Net	M	[64]	2018
	LSTM+CNN	INRIX data Seattle	3.28		3.74		SGC-LSTM	L	[84]	2018
	LSTM	AIMSUN intersection		RMSE:7.931(V.sec)		RMSE:8.586	Kalman Filter	L	[85]	2018
	CNN	NGSIM	3.84		8.56		SVR	S	[50]	2019
	CNN	PeMS	0.105		0.124		GCNN	S,M	[86]	2019
	DNN	Alberta Intersections		R-squared:0.89					[87]	2019
	GAN	PeMS	36.91		37.63		SAE	S	[88]	2019
	SAE	Beijing Roads		RMSE:18.30		RMSE:20.8	LSTM	S	[89]	2019
	SAE+DNN	Beijing Car	12		15		BPNN	S	[90]	2019
Traffic Conditions	RBM+RNN	Ningbo, China		AC:88.2%		AC:71%	SVM	L	[58]	2015
	CNN	VPRs Jinan, China		RMSE:0.241		RMSE:0.285	LSTM	S,M,L	[91]	2018
	DLN+CNN	Motorway dataset,UK		AC:89%		AC:70.5%	AE		[92]	2018
	YOLO	IOWA DOT CCTV		AC:91.4%		AC:85.7%	SVM		[93]	2018
	DBN+SVR	PEK airport data		RMSE:12.65(min)		RMSE:16.01	KNN		[66]	2019
	CNN+RNN	Taxi-Schenzan data		AC:97.59%		AC:77.92%	GRU	S	[94]	2019
Traffic density	CNN	Seoul Intersection		RMSE:4.15(car count)		RMSE:7.109	Crowd CNN		[95]	2018
	CNN	Traffic Simulator	3.68		9.34		ARIMA	S	[50]	2019

4.3. Traffic incidents inference

The goals of predicting traffic incident risk for a given location as well as incident detection based on traffic features are to help traffic management agencies to reduce incident risk in a hazardous area and traffic jams in incident locations. Although there are parameters such as drivers' behavior, that are not very predictable, there are several key features that can help predict traffic incidents.

Table 2: Overview of papers using deep learning techniques for traffic incident inference.

Characteristic	Model	Dataset	Experiment Results		Baseline Results			Paper	Year
			RMSE	Others	RMSE	Others	Model		
Incident risk	SdAE	7 month heterogeneous data	1		1.41		Logistic Regr.	[96]	2016
	LSTM	Accident records(Beijing)	0.63		0.75		SdAE	[97]	2017
	LSTM	NYPD	9.44		10.46		CNN	[98]	2019
	DNN	Accident records(VTTI)		AC:85%		AC:69%	Decision Tree(DT)	[99]	2019
	DBN	Civil Aviation Data		MSE:0.2		MSE:0.05	SVR	[100]	2019
Traffic incident prediction	DNN	Accident records(IOWA)		AC:95.12%		AC:89.58%	RF	[101]	2017
	DBN	Highways dataset	1.48		1.60		Bayesian ANN	[102]	2017
	AE	TRIMS and PMS		MAE:0.150		MAE:0.660	SVM	[103]	2018
	LSTM	Accident records(IOWA)	0.078		0.121		7 year Avg.	[104]	2018
Collision prediction	DNN	Internet of vehicles						[105]	2018
	DBN	Collision Data, Ontario	15.24		16.51		Bayesian NN	[106]	2018
	LSTM	Annotated data		mAP:62.1%		mAP:57.8%	RNN	[107]	2018
	DNN	Attica Tollway		AC:68.95%		AC:72.15%	DT	[108]	2019
Incident detection	SAE	SUMO		MSE:0.13		MSE:0.18	DNN	[109]	2018
	DBN	NYC+NOVA Accidents		AC:85%		AC:79%	SVM	[110]	2018
Incident severity prediction	LSTM	Accident records(Malaysia)		AC:71.77%		AC:70.30%	Bayesian LR	[111]	2017
	DNN	Accident records(AbuDhabi)		AC:74.6%		AC:59.5%	Ordered Probit	[112]	2017
	CNN	Accident records(Louisiana)	0.231					[113]	2018

Human mobility [96], traffic flow, geographical position, weather, time period and day of the week [97] are some of these features that can be investigated as indicators of a traffic incident. However, a single model cannot generally be used in different places because accident factors in metropolitan areas, where the population and vehicles are generally dense, are completely different from accident factors in a small town with a scattered population [101]. The prediction and detection of an incident is more challenging than the prediction of incident risk since data for the former are usually heterogeneous (i.e, traffic incidents happen rarely, compared to the amount of data for the cases where there is no incident). To overcome this issue, [101] in each step changed only one feature of the data (hour, day, or location) and then checked if the resulting data point was negative or not. In negative cases, it was added to the pool of data to be considered.

To measure the traffic incident risk based on surveillance camera data, different approaches have been used. For example, [96] have used a stack denoising AE (SDAE) to learn the hierarchical features of human mobility and their correlation with a traffic incident. In contrast, [97, 98] have implemented an LSTM model to evaluate risk, but [97] achieved better performance due to learning from more features.

To predict traffic incidents in a macroscopic manner, [101, 102] have tried implementing DNN models, [101] by considering the curvature of the road as well as the number of intersections and density of the area in order to overcome the spatial heterogeneity problem. For the same concern, [103] have used AE by considering both continuous and categorical variables, and [104] have used a Conv-LSTM that breaks regions into smaller regions in order to overcome spatial heterogeneity.

If, following [104], we consider the macroscopic prediction of traffic incidents as not focused on any single vehicle, but instead as predicting the probability of an accident between any pair of vehicles in the wider region, microscopic incident prediction studies can also be introduced that—by getting data about the location, speed and direction of each vehicle in the surrounding area—predict the probability of an incident in the near future between any certain pair of vehicles. In this regard, [105, 108] have trained a DNN to predict likely collisions. [108] have used a simple NN with 4 layers, which, though it does not compare well with the baseline results of machine learning (ML) techniques, is still preferred, as the ML techniques have poor sensitivities.

[107] have annotated their large dataset of near-miss traffic accidents to train a quasi-RNN model. The innovation of their work was introducing an adaptive loss function for early anticipation (AdaLEA), which gives their model the ability to predict a collision 3.65 seconds before it happens.

Another challenge in traffic incident inferencing is detecting an accident by processing only raw data. To address this, [109, 114] have used a stacked AE (SAE) to extract the features of traffic patterns in the context of an accident. Also, [109] have used a fuzzy DNN to control the learning of traffic-incident-related parameters. [110] have trained their DBN model on a dataset that includes tweets related to traffic accidents, showing that non-traffic features can be used along with traffic feature data to validate traffic incident detection.

Incident severity prediction based on recorded incident features have been studied in [111, 112, 60]. The artificial neural network (ANN) trained in [112] has shown an improvement in baseline performance as compared to the LSTM model with fully connected layers in [111].

Table 2 summarizes all these papers, shows their model, the dataset which their model was trained on, evaluation of their model for their testing dataset as well as comparison of their model's performance to that of their baseline model. In the first section of this table, different studies regarding parameters effective in predicting increased incident risk and the manner in which incident risk is affected are listed. In the next section, macroscopic studies on incident prediction are categorized as "traffic incident prediction," whereas microscopic studies are categorized as "collision prediction." In the incident detection section, all studies focused on detecting incidents by analyzing raw traffic data have been gathered and, finally, in the last section, investigations predicting the severity of the incident are listed.

4.4. Vehicle identification

Applications of re-identification (Re-ID) vary from calculating travel time to automatic ticketing. Since license plates are unique to each vehicle, the first task in Re-ID is recognizing them.

[115, 116] have implemented DL models to recognize license plates by using a visual attention model that first generates a feature map using a combination of the most commonly used colors in license plates, extracts data from plates using a CNN model, and ultimately runs an SVM on the extracted data. However, bad lighting, blurriness due to vehicle movement, low camera quality and even traffic occlusion where the plate is covered behind other cars can make reading license plate characters impossible. To overcome this, [117] have proposed a CNN layer to extract conspicuous features such as the color and model of the vehicle and have used a Siamese neural network to distinguish similar plates. (This network has been used before in signature verification tasks.) Note that for some feature extractions, such as vehicle color recognition, solutions like what [118] did using a combination of CNN for feature extraction and SVM for categorizing are also available. [119] have similarly used a histogram-based adaptive appearance model like what [120] did for target re-identification, detecting and saving other features of each car besides the scheme of the license plate to do Re-ID. Also, [121] have used faster RCNN to detect vehicles in images. In addition, a modified version of the Single Shot Detection (SSD) method to localize and classify the different types of construction equipment by employing MobileNet as the feature extraction network has been done by [122]. [123] has worked on the same idea but trained their model based more on spatiotemporal data, pruning their results with the fact that 1) a vehicle cannot be in two places at one time and 2) a vehicle that has already passed a section is unlikely to pass it again. However, their model could not compete with the model defined in [119], that proposed a Markov chain random fields to prepare several queries based on a visual spatiotemporal path and then used a combined Siamese-CNN and path-LSTM model.

Table 3 summarizes all these papers, show their models, the dataset which model is trained on and their performances on those dataset and comparison to the baseline model.

Table 3: Overview of papers using deep learning techniques for vehicle id tasks.

Characteristic	Model	Dataset	Experiment Results		Baseline Results			Paper	Year
			AC%	Others	AC%	Others	Model		
License plate recognition	SIFT+SVM	Chinese license plate		PR:98.6		PR:98.4	CNN+SVM	[115]	2015
	CNN	On road data	99					[124]	2017
	CNN	Bangla license plate	92					[116]	2018
	Kernel-CNN	Chinese license data	96.38		93.35		SVM-RBF Kernel	[125]	2018
	CNN	Thailand license plate	96.94					[126]	2018
	CNN	AOLP		PR:99.5		PR:90.7	Single shot detector	[127]	2018
Vehicle type classification	CNN	Generated dataset	99.07					[128]	2015
	CNN	BIT-vehicle dataset	96.1			93.7	SVM	[129]	2015
	Faster RCNN	Vehicle dataset	89					[121]	2017
	CNN	VEDAI/Munich dataset	54.6/73.7		32/53.9		Fast RCNN	[130]	2017
	CNN	ILSVRC2012	98.29		83.78		Ensemble classifiers	[131]	2017
	CNN	Iowa CCTV data		PR:95				[132]	2017
	Deep CNN	CarFlag/CompCars	98.67/99.3		90.34/93.58			[133]	2017
	Deep CNN	XMUPlus	99.10		99.07		Pretrained CNN	[134]	2018
	Deep CNN	MIT-CBCL/Caltech	94.12/95.04		93.71/94.27		RCNN	[135]	2018
	CNN	COSMO-SkyMed	97.66		95.48		CNN	[136]	2018
	CNN	CompCars	54.56		42		Pre-trained CNN	[137]	2018
	Faster RCNN+RPN	ILSVRC-2012		mAP:89.93%		mAP:89.12%	Faster RCNN	[138]	2018
	LSTM	Fleetmatics data, US	85					[139]	2018
	D-CNN	CompCars		Top 5:0.922		Top 5:0.917	CNN	[140]	2018
	Fast RCNN+RPN	MIT/CALTECH dataset	84.4		84		Fast RCNN	[141]	2018
	Deep-CNN+AE	Chengdu expressways	97.62		95.18		CNN	[142]	2018
	CNN	LabelMe/BIT datasets	98.95/95.12					[143]	2019
	PCA-CNN	PLUS Malaysia NSE	99.51		98.65		Ensemble Classifier	[144]	2019
RE ID	CNN+SNN	VeRi-776		mAP:27.77%		mAP:18.49%	CNN	[117]	2016
	MRF+SNN+LSTM	VeRi-776		mAP:58.27%		mAP:46.25%	LSTM+CNN	[145]	2017
	CNN	VOT2016	54		54		CNN	[146]	2017
	CNN	AI City Challenge		PR:99.25				[119]	2018
	CNN+AFL	VeRi AI City Challenge		mAP:57.43%		mAP:58.27%	CNN	[123]	2018
	CNN	AI City Challenge		PR:99.25				[147]	2018
	RNN	Brisbane vehicle data		PR:37.5				[148]	2018
	Faster RCNN	Korea Highways		MAPE:3.4%		MAPE:4.9%	SSD	[149]	2019
Vehicle color recognition	CNN SVM	Vehicle Color Dataset	93.78		91.89		SVM	[118]	2015

4.5. *Traffic signal timing*

One of the main tasks of ITS management based on multiple types of data is controlling traffic via traffic signal lights. For several years, research on optimizing signal light timing to have the best performance has been one of the greatest challenges in the transportation field. The results of studies in this area have endowed traffic agencies with analytical models that use mathematical methods to address this optimization problem. However, through emerging DL studies, modeling the dynamics of traffic to achieve the best performance has taken a new path. This is because the nature of RL has facilitated its application in different studies to find the best traffic signal timing.

[150] has used DRL to tackle traffic light timing. In DRL, a DL model is usually used to implement the Q-function in a complex system to capture the dynamics of traffic flow. A dSAE network is used to take the state as input and give the Q-function for any possible action as the output of the network. [150] has shown a 14% reduction in cumulative delay in the case of using an SAE to predict the Q-function instead of conventional prediction.

[151] has suggested an alternative novel idea for choosing RL states. They argue that instead of taking raw data as the state, it could be more effective if the CNN extracts important features from the raw data—e.g., the position of the cars and their speeds—and feeds it to a DRL network with a fully connected network to predict the Q-value for each of 4 states of green, yellow, red and protected left turn light, considering cumulative staying time as the reward. They have also used the experience replay and target network techniques to stabilize the algorithm and converge it to the optimal policy as suggested in [27].

[152] have also used CNN to map states. They use several state-of-the-art techniques such as the target network, experience replay, double Q-learning network and dueling network methods to increase the performance of the network and make it stable. Their results have shown a great reduction in waiting time (more than 30%) for a fixed-time scenario.

[153] have investigated the importance of choosing delay time states. The main goal of this study was investigating whether the data from conventional sensors, such as occupancy and average speed, are satisfactory or more precise data are needed, such as vehicle density and queue length, or even data with the highest resolution, such as discretizing each incoming lane into cells and considering the presence of a vehicle in each cell separately. The results of this study showed that using high resolution data is not substantially effective and conventional data are good enough for their model. However, one of the reasons that may have contributed to this conclusion is that they used a simple fully connected model that could not extract deep features from more precise states very well.

Finally, [154] have tested their model on real-world traffic data to see how effective its results could be. They suggest that instead of only studying the reward, we need to consider different policies that may result in the same reward and then take the most feasible one. The final results of this study have shown great performance in reducing queue length, delay time and duration compared with other methods.

Table 4 summarizes all these papers, shows their model, the dataset which their model was trained on, and the performance of their model for the testing dataset as well as comparison of their model's performance to that of the baseline model.

Table 4: Overview of papers using deep learning techniques for traffic signal timing.

Paper	Year	Model	Dataset	State	Reward	Actions
[150]	2016	SAE+DRL	PARAMICS	queue length	—Queue length difference—	2
[155]	2016	DQN	SUMO	position, speed	number of Stop switch and delay	2
[151]	2017	CNN+DRL	SUMO	F(position, Speed)	(cumulative staying time)	4
[152]	2017	CNN+DRL	SUMO	position, speed	(cumulative waiting time)	8
[156]	2017	DQN	SUMO	snapshot of the current state of a graphical view of the intersection	difference between the total cumulative delays of two consecutive actions	2
[153]	2018	DRL	SUMO	occupancy and speed/vehicle density and queue length/presence of vehicles in each lane	change in cumulative delay	4
[154]	2018	DRL	China	queue length, number of vehicles, updated waiting time, current phase, next phase and an image of the intersection analyzed by CNN	F(queue length, delay, updated waiting time, light switches indicator, number of vehicles pass the intersection, travel time)	2
[157]	2018	DQN	VISSIM	current phase, green and red duration, remaining cars and left turn bay occupation	system delay	8
[158]	2018	DRL	VISSIM	queue length, signal state, and time of day	Discharged vehicle	2
[159]	2019	3DQN	SUMO	the position and speed of vehicles	change of the cumulative waiting time between two neighboring cycles	8
[160]	2019	3DQN	Simulated traffic	current traffic state and current signal phase	difference between the current and previous waiting times of all vehicles	4
[161]	2019	DQN	SUMO	number of input and output vehicles of adjucents intersections	summation of que length in multiple intersections	4

4.6. *Ride sharing and public transportation*

Public transportation systems (including bus or metro systems, taxis, etc.) are one of the main means of moving passengers within cities. To increase city planning performance and also passenger satisfaction, the nature of DNN has endowed companies with increasingly optimal routing maps that take into account data such as passenger demand for a given mode of travel at particular places and times. DL has been adopted to make predictions even more accurate compared to existing ML techniques.

[162] have investigated the performance of several ML techniques and a fully connected DL model with only 2 hidden layers and have shown that their very simple DL model outperforms almost all other techniques except a boosted decision tree. Besides the simple DNN models in [163, 164, 165, 166], a hybrid model containing a stacked AE and a DNN has been implemented by [167] to predict hourly passenger flow.

To capture all related features such as the spatial, temporal and exogenous features impacting passenger demand, a fusion convolutional LSTM network (FCL-Net [168]) has been proposed. This network includes stacked Conv-LSTM layers to analyze spatiotemporal variables, such as historical demand intensity and travel time, and LSTM layers to evaluate nonspatial time-series variables, such as weather, day of the week and time of day. With the same idea, [169] has proposed a spatiotemporal Resnet (ST-Resnet) which includes several convolutional layers. [170] has implemented both of these techniques on a New York City taxi record dataset and their comparison has shown that better performance with a faster training time can be achieved using ST-Resnet. The authors suggest two reasons for this. First, LSTM captures fine temporal dependencies which are not as fundamental as the coarse-grained dependencies from the convolutional layers. Their second explanation is that spatial features may be more important than temporal ones and since the ST-Resnet focuses more on spatial features, it outperforms the FCL-Net. [120, 171] work directly on graphs structures to leverage structural information by considering the nodes as stations and the edges as dependencies among stations. Finally, [172, 173] have proposed a deep multiview spatiotemporal network to capture all dependencies separately.

Another research area related to public transportation deals with travel mode selection. [174] has implemented a simple fully connected DNN on Swiss Metro data to reveal demand based on mode. Another issue for transportation network companies is route scheduling for their drivers to pick up passengers in order to minimize passenger waiting time as well as cost for the driver and company. [175] has suggested a DRL model aiming to give drivers the best route. This paper considers different factors such as the current location of vehicles, time of day, and competition between drivers, resulting in a significantly shorter search time and more long-term revenue for drivers.

Table 5 summarizes all these papers, shows their model, the dataset which their model was trained on, and evaluation of their model for their testing dataset as well as comparison of their model's performance to that of their baseline model. (In this table, "travel mode" refers to studies which tried to predict the mode of transportation that passengers would choose at each time point. Also, "passenger flow" is defined as the number of passengers flowing in or out of a given location at a certain time point.)

Table 5: Overview of papers using deep learning techniques for ride sharing and public transportation.

Characteristic	Model	Dataset	Experiment Results		Baseline Results			Paper	Year
			RMSE	Others	RMSE	Others			
Travel mode	DNN	Swiss Metro dataset		AC:66.1%		AC:65.57%	ANN	[174]	2017
	CNN	GPS - GeoLife project		AC:84.8%		AC:78.1%	RF	[176]	2018
Route scheduling	DRL	Didi Chuxing						[175]	2019
Passenger flow	SAE+DNN	Xiamen bus station	50.4		51.4		SVM	[167]	2017
	CNN	Passenger data		AC:96%				[163]	2017
	CNN	AFC, Seoul		AC:60.10%		AC:54.83%	Statistics	[164]	2017
	DNN	Unity-3D environment						[165]	2018
	DNN	Shanghai rail transit		MSRE:0.00000125		MSRE:0.00178	Lin. Regr.	[166]	2018
	CNN	California HTS		AC:93.59%		AC:69.76%	RF	[177]	2018
	LSTM	Nanjing Metro System	8.19		11.54		ARIMA	[178]	2019
	AE+LSTM	Singapores Metro System	20.37		24.82		LSTM	[179]	2019
Passenger demand prediction	ConvLSTM	Didi Chuxing	0.016		0.0175		CNN	[168]	2017
	CNN	Beijing taxi/NY bike	16.69/6.33		18.18/7.43		DNN	[169]	2017
	DNN	Didi Chuxing	20.09		16.41		DT	[162]	2017
	CNN+LSTM	Didi Chuxing	9.642		10.012		XGBoost	[172]	2018
	GCNN-DDGF	Citi Bike data, NY	2.12		2.43		XGBoost	[171]	2018
	LSTM	TAZ Nanjing,China		MAPE:46.49%		MAPE:65.128%	XGBoost	[180]	2018
	CNN+RNN	Porto Taxi Trajectory		AC:78.80%		AC:75.62%	CNN	[181]	2018
	DQN	London travel data		Waiting time:158.2				[182]	2018
	CNN	Citi Bike System, NY	18.995		19.784		NN	[183]	2018
	DNN	NYC taxi data	11.13		16.05		LSBoost	[170]	2018
	CNN+LSTM	Beijing metro	7.5		8.89		LSTM	[173]	2019
	GCN	Seouls Bike data	2.26		2.45		LSTM	[184]	2019

4.7. *Visual recognition tasks*

One of the most significant applications of DL is the use of nonintrusive recognition and detection systems, such as camera-image-based systems. These applications can vary from providing a suitable roadway infrastructure for driving vehicles to endowing the autonomous vehicles with a safe and reliable driving strategy.

One of the first visual recognition challenges tackled has been obstacle detection via exploiting vehicle sensors. To do this, a variety of networks with unique architectures have been implemented. [185] have merged data from an RGB camera and LIDAR sensors to increase obstacle detection performance in different illumination conditions. [186, 187], on the other hand, have confronted obstacle detection as an anomaly detection problem. They have used a hybrid encoder model to extract features of Deep Boltzmann Machine (DBM) and then an autoencoder to reduce the dimensionality and obtain vertical disparity (V-disparity) map coordinate system data from images. The key feature of V-disparity data is that these data are mostly stable with small variations from noise and they change drastically only if an obstacle appears in an image.

[188, 189] have used data from far-infrared sensors to improve vehicle detection at night. While the former used only far-infrared data, the latter, in order to decrease the false positive percentage used both camera and far-infrared data. [190] have tried to address requirements in regard to vehicle following, which include detecting brake lights. They used the Histogram of Oriented Gradient (HOG) approach implemented with LIDAR and camera data. To decrease the false positive rate and speed up the process, they also used the vanishing point technique. Next, they used AlexNet to detect if the rear middle brake light was on or off.

Another important task in navigating safely is traffic sign detection. These signs obligate, prohibit or alert drivers. One of the most common DL models to detect traffic signs are CNNs. [191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203] have all used CNN as their main feature extractor, each trying to tune their model to get the best results. [191] have used RCNN to derive regions of interest from RGB images. [194] have focused on low-illumination images. They used a classifier to detect regions of interest and an SVM to verify if any traffic signs were present inside the region or not. Then, a CNN model using the Byte-MCT technique classified the traffic sign. Experiments have shown that this method is robust in deficient lighting, outperforming other methods in cases of low illumination.

[195] have suggested that the RGB space cannot provide as much useful data as the perceptual lab color space. Therefore, after space changing, they extracted the deep perceptual features using a CNN and fed these features to a kernel-based ELM classifier to identify the traffic sign. This classifier used the radial basis function to map the features in a higher dimension space in order to disconnect features to get the best outcome.

[198] have tried different optimization methods on a CNN model containing several convolutional layers and spatial transformer networks (STN) that make the CNN spatially independent, resulting in no need for supervised training, data augmentation or even normalization. In contrast, [204], instead of using a CNN, have used a DBM that is boosted with canonical correlation analysis for feature extraction and then an SVM for classification. Also, they have used certain conventional image-processing techniques such as image drizzling and gray-scale normalization to reduce noise.

[205, 206, 207] have focused more on traffic light detection and classification. This has a very significant role in managing traffic, and correct detection has a high correlation to reduced risk. [205] have proposed their Deep Traffic Light Recognition (DeepTLR) model that first classifies each fine-grained pixel of the input data, calculating the probability for each class. Then, for the regions with higher probability toward the presence of a traffic light, a CNN was used to classify the status of the traffic light. (In this model, temporal data was not used and each frame was analyzed separately.) However, [206] have used traffic speed information as well as stereovision data to track detected traffic lights. [193] have used a combination of region-of-interest (ROI) performance, CNN feature extraction and an SVM as a classifier to detect arrow signs on the roadway and classify their direction. [208] have used a CNN to detect lane position in the road.

Finally, the monitoring of civil infrastructure has always been a focus for engineers and researchers. Various monitoring techniques have been used for infrastructure performance evaluation, ranging from conventional short-term [209] and long-term [210, 211, 212] sensor-based monitoring to nondestructive and noncontact techniques [213]. Among the applications of nondestructive damage detection, pavement crack detection, in particular, has received attention, due to its importance in civil infrastructure management. For instance, [214] have proposed a unified pavement crack detection approach that can distinguish between cracks, sealed cracks, and background regions. Through their approach, they have been able to effectively separate different cracks having similar intensity and width. Moreover, [215] have proposed pixel-level pavement crack detection in black-box images using an encoder-decoder network and found that ResNet-152 with transfer learning outperformed other networks. Additionally, CrackNet, which performs pixel-level pavement crack detection on laser-based 3D asphalt images, was introduced by [216]. In a separate study, [216] extended their previous study to CrackNet-R, which utilizes RNN with a gated recurrent multilayer perceptron (GRMLP) to update the memory of the network, showing their model outperforms other models based on LSTM and GRU. Also, [217] have investigated pavement crack detection performance using metaheuristic-optimized Canny and Sobel edge detection algorithms, comparing these algorithms with their proposed CNN and confirming the superior performance of DL over conventional edge detection models.

Table 6 summarizes all these papers, shows their model, the dataset which their model was trained on, and evaluation of their model for their testing dataset as well as comparison of their model's performance to that of the baseline model.

Table 6: Overview of papers using deep learning techniques for visual recognition tasks.

Characteristic	Model	Dataset	Experiment Results		Baseline Results			Paper	Year
			AC%	Others	AC%	Others	Model		
Obstacle Detection	Fast RCNN	KITTI		PR:88.99		PR:88.01	CaffeNet	[185]	2016
	AE CNN	CCD Stereo data	98.15		96.14		RCNN	[218]	2016
	CNN	Caltech Pedestrian		Missrate:54%		Missrate:69	HOG+SVM	[219]	2017
	SdAE+KNN	Bahnhof data	91		81		DBN	[186]	2018
	AE+SVM	Malaga, Daimler data	93.08		89.53		SVM	[187]	2018
	CNN	Video data		PR:95		PR:90	CNN	[220]	2018
	CNN	Video data	96.8					[221]	2018
	CNN	GMVRT/ UCF-ARG	99.71		92.36		HOG+SVM	[222]	2018
	CNN	Railway video data		mAP:89.53%		mAP:88.61%	SSD	[223]	2018
	CNN	Caltech data		Missrate:42.27		Missrate:60.95	MS-CNN	[224]	2018
	CNN	FCTD Camera		PR:90.81		PR:70.61	SSD	[225]	2019
	DNN	Video data	98					[226]	2019
Vehicle detection	DBN	Far Infrared images		RL:93.9		RL:91.4	SVM	[188]	2016
	DBN	Far Infrared images		RL:92.3		RL:91.8	DBN	[189]	2016
	CNN	Built from videos		Recognition rate: 94.68				[227]	2017
	HRPN+Boost Classifiers	Munich vehicle dataset		PR:89.2		PR:86.2	HRPN	[228]	2017
	Deep CNN	Recorded vehicle data		Top 5:97.51%				[229]	2017
	DNN	LISA 2010		PR:81.10		PR:77.09	Faster RCNN	[230]	2018
	DBN RNN	KITTI	95.36		92.82		Encoded SVM	[231]	2018
	Scale-Insensitive CNN	KITTI	89.60		89.02		MS-CNN	[232]	2018
	CNN	Video data	90.7		90.4		CNN(Resnet)	[233]	2019
Traffic sign recognition	DBN	GTSRB	96.68		95.16		HOG	[204]	2016
	HOG+DBM	GTSDB	96.68		95.16		HOG	[192]	2016
	CNN+SVM	Built from videos	71.87					[193]	2016
	CNN+SVM	Korea daylight		PR:99.03		PR:73.49	CNN	[194]	2017
	CNN+KELM	GTSRB	99.54		99.65		Ensemble CNN	[195]	2017
	CNN	GTSDB	99.4		77.3		HOG	[234]	2017
	Fast BCNN	GTSRB	99.01		99.12		BCNN	[196]	2017
	CNN	MASTIF	97.78		98.97		R-LSTM	[197]	2017
	CNN+STN	GTSRB	99.71		99.65		Ensemble CNN	[198]	2018
	CNN	GTSRB	99.75		99.67		CNN	[199]	2018
	CNN	SDTS		PR:89.4				[200]	2018
	CNN	GTSDB		PR:90.7		PR:84.20	HOG+SVM	[201]	2019
	CNN	HDR		PR:94.24		PR:89.33	Guassian Mixture	[202]	2019
	CNN	GTSRB		mAP:83.3%		mAP:80.8%	CNN	[203]	2019
Traffic light recognition	CNN	LaRA data		PR:96.9		PR:61.22%	Image Proc.	[205]	2016
	CNN	Bosch Traffic Lights	95.1					[206]	2017
	Faster RCNN	Bosch Traffic Lights		mAP:20.40%				[207]	2018
Lane detection	DNN	Generated data		Top 5:98.55%				[208]	2016
	CNN	Caltech	99.35		97.21		Image Proc.	[235]	2018
	GBNN	NGSIM	97.7		96.6		CNN	[236]	2019
	CNN		98.37					[237]	2019
Vehicle signal detection	HOG CNN	Built from videos	99					[190]	2016
	FRCN+RPN+F	SYSU data	95.58		94.61		FRCN+RPN	[238]	2017
Road surface detection	RNN+LSTM	Built from videos	94.6					[239]	2018
	Deep CNN	Cambridge	100		82.6		Faster RCNN	[240]	2019
Street scene labelling	S-CNN	Camvid	53.2		47 .4		FCN	[241]	2018
Traffic scene segmentation	AE		78.8		76.4		SegNet	[242]	2018
	CNN+IAL	Cityscape		IoU:74.8		IoU:71.3	CNN	[243]	2019
	CNN+MFI	Built from videos	91.7		81.1		CNN	[244]	2019
Crack detection	CNN	3D pavement data	94.29					[245]	2017
	CNN	PaveVison3D data		PR:90.20		PR:90.13	CrackNet	[216]	2018
	RCNN	Cifar-10 data						[246]	2018
	Deep CNN	Da Nang, Vietnam	92.08		81		DFP-Sobel	[217]	2018
	Deep CNN	Captured Data		PR:84.7		PR:51.5	RF	[214]	2018
	Deep CNN	Railway data	97.8					[247]	2019
	Deep CNN	Generated data, Seoul		PR:77.68		PR:25.14	SegNet	[215]	2019

Table 7: Detailed specifications of the popular edge computing devices used for DL.

	Jetson Xavier	Jetson TX2	Jetson Nano	Raspberry Pi 3 B+	Intel NCS 2
GPU	512-core Volta GPU @ with 64 Tensor Cores	NVIDIA Pascal, 256 CUDA cores	128-core Maxwell	Broadcom VideoCore IV	Intel® Movidius^{TM} Myriad^{TM} X VPU
CPU	Octal-core NVIDIA Carmel ARMv8.2 CPU @ 2.26GHz	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2	Quad-core ARM A57 @ 1.43 GHz	4* ARM Cortex-A53, 1.2GHz	N.A.
Memory	16 GB 256 bit LPDDR4 137GB/s	8 GB 128 bit LPDDR4 59.7 GB/s	4 GB 64-bit LPDDR4 25.6 GB/s	1GB LPDDR2 (900 MHz)	N.A.
Display	3x eDP 1.4, DP 1.2, HDMI 2.0	2x DSI, 2x DP 1.2, HDMI 2.0, eDP 1.4	HDMI 2.0, eDP 1.4	HDMI, DSI	N.A.
Data storage	32GB eMMC 5.1	32 GB eMMC, SDIO, SATA	microSD	microSD	N.A.
USB	USB C	USB 3, USB 2	USB 3, USB 2	USB 2	N.A.
Connectivity	1 Gigabit Ethernet	1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth	Gigabit Ethernet	100 Base Ethernet, 2.4GHz 802.11n wireless	USB 3
Mechanical	105 mm x 105 mm	50 mm x 87 mm	100 mm x 80 mm	56.5 mm x 85.60 mm	72.5 mm x 27 mm
Power	10 W, 15 W, 30 W	7.5 W	5 – 10 W	5 W	1 W
Price	1299 USD	599 USD	99 USD	35 USD	99 USD

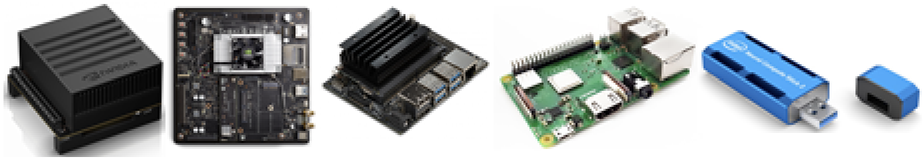


Figure 4: Hardware: NVIDIA Jetson Xavier, NVIDIA Jetson TX2, NVIDIA Jetson Nano, Raspberry, Intel NCS 2

5. Discussion and Conclusion

5.1. Hardware

Generally, there are two types of intelligent decision-making, namely cloud-computing-based and edge-computing-based. While computing services are delivered over the internet via the cloud computing approach, they are performed at the edge of the network via the edge computing approach. The edge computing approach has introduced several advantages, such as efficient and fast intelligent decision-making as well as decreased data transfer cost. Emerging technologies such as DL have significantly increased the importance of edge computing devices. Though discussing edge computing devices in detail goes beyond the scope of this paper, we briefly overview and compare the edge computing devices popularly used for DL. Figure 5.1 illustrates the various edge computing platforms discussed in this section. Also, Table 7 summarizes the technical specifications of the covered hardware.

The Jetson Xavier is the high-end system-on-a-chip (SoC) computing unit in the Jetson family, which exploits the Volta GPU. An integrated GPU with Tensor Cores and dual Deep Learning Accelerators (DLAs) make this module ideal to deploy computationally extensive DL based solutions. NVIDIA Jetson Xavier is capable of providing 32 TeraOPS of computing performance with a configurable power consumption of 10, 15 or 30W.

Another widely used embedded SoC is NVIDIA Jetson TX2 which takes advantage of NVIDIA Pascal GPU. Although it delivers less computing performance than NVIDIA Xavier, it can be a reliable edge computing device for certain applications. The module can provide more than 1TFLOPS of FP16 computing performance using less than 7.5W of power consumption. The Jetson Nano, which utilizes the Maxwell GPU, is newest product from the Jetson family introduced by NVIDIA. It is suitable for deploying computer vision and other DL models and can deliver 472 GFLOPS of FP16 computing performance with 5-10W of power consumption.

Another family of edge computing devices is the Raspberry Pi family, which introduces affordable SoCs capable of high performance in basic computer tasks. The Raspberry Pi3 Model B+ is the latest version of the Raspberry Pi which uses a 1.4GHz 64-bit quad-core processor and can be used alongside deep learning accelerators to achieve high performance in computationally expensive tasks.

Finally, the Intel Neural Computing Stick 2 (NCS 2) is a USB-sized fanless unit, which utilizes the Myriad X Vision Processing Unit (VPU) that is capable of accelerating computationally intensive inference on the edge. Very low power consumption along with supporting popular DL frameworks such as Tensorflow and Caffe have made the NCS 2 ideal to use with resource-restricted platforms such as Raspberry Pi3 B+. There have been limited studies investigating the inference speed of these hardware, though [122] has compared the inference speed of an SSD-MobileNet model of the abovementioned embedded devices on a construction vehicle dataset. Utilizing the Jetson TX2, they achieved 47 FPS, and utilizing a Raspberry Pi and NCS combination, they achieved 8 FPS.

5.2. Summary

Below, we provide a summary of the studies cited in the current paper. We have classified these studies according to our six ITS application categories in relation to the DL models they use 5. The following are our observations

- Traffic Characteristics: CNN, RNN, and CNN-RNN hybrid models are most frequently used. The main reason is undoubtedly related to the nature of traffic that has two main dependencies: spatial and temporal. Because various datasets and performance evaluation metrics have been used, it is hard to compare different studies related to traffic characteristics, but in traffic flow studies, the PeMS dataset has been widely used. The majority of research has used hybrid CNN and RNN models, which can identify both long temporal dependencies and local trend features. Although most papers have defined their own CNN model rather than using an existing architecture, CNN has generally shown better performance across papers when compared to RNN, which shows lower computation/training time.

- Traffic Incidents: The most widely used model is RNN, since the result of an incident shows itself at a specific time that requires a powerful network model to identify. Autoencoders are also popular models, since they can learn traffic patterns and then detect and isolate accident conditions from regular conditions.

- Vehicle ID: CNN is the most widely used model, given its power in inferencing from images, as detection and tracking is the main task in license plate and vehicle type/color identification. Existing CNN architectures that have been popularly utilized are AlexNet and VGG models that have been pretrained on ImageNet.

- Traffic Signal Timing: RL has been the most commonly used model, given the control strategy nature of the traffic signal timing task. Hybrids of CNN and SAE have been used to approximate or learn Q-values to improve DRL performance.

- Ride-sharing and Public Transportation: CNN, RNN, and DNN have been the most frequently used models in the domain. Most researchers have built their own DL architecture to accomplish tasks in this category. Public transportation demand and traffic flow prediction tasks have generally been done by either CNN or hybrid CNN models.

- Visual recognition tasks: CNN has been the most commonly used DL model for visual recognition tasks, again because detection and tracking are efficient via CNN. Especially in traffic sign recognition tasks, the GTSRB dataset has been one of the most frequently used benchmarks. Existing architecture such as ResNet, AlexNet, VGG and YOLO have been used extensively, with the AlexNet and ResNet architectures being the most popular to build on. This can be attributed to the fact that visual recognition tasks are not limited to ITS, so research done in other domains can be utilized to accomplish ITS-related visual recognition tasks. .

Based on all the studies reviewed in the current paper, deep learning as an approach for addressing intelligent transportation problems has undeniably achieved better results as compared to existing techniques. The major growth has been seen in the last 3 years, constituting more than 70% of all ITS-related DL research performed so far.

5.3. Future work and Challenges

In recent years, DL methods have been able to achieve state-of-the-art results in different visual recognition and traffic state prediction tasks. The majority of the visual recognition work such as vehicle and pedestrian detection, traffic sign recognition have focused on autonomous driving or in-vehicle cameras. However, there have also been a significant number of overhead cameras installed by city traffic agencies and state Departments of Transportation, that are mostly used for human-evaluated surveillance purposes. To date, there have been only a few studies that have focused on using these cameras for determining traffic volumes on freeways and arterials, traffic speed, and also for surveillance purposes such as automatically detecting anomalies or traffic incidents (particularly at a large-scale, citywide level). Currently, the majority of traffic intersections rely on using loop detectors for vehicle counting and for developing actuated traffic signals. However, installation of these loop detectors is intrusive, in that road closures are required for installing such sensors. Cameras, on the other hand, can be used as a cheap, nonintrusive detection sensor technology for counting traffic volume in all directions as well as turning movements, the presence of pedestrians, etc., thereby facilitating smart traffic signal control strategies. However, two main challenges need to be considered for developing DL techniques able to handle the use of cameras as sensors. Firstly, such methods need to be able to handle the large volume of data collected from hundreds or thousands of cameras installed at a citywide or statewide level. Efficiently providing real-time or near-real-time inferencing from this large volume of data is currently one of the primary challenges of using cameras as sensors. Secondly, the methods developed need to be able to perform with minimal or no calibration such that they are feasible to apply and maintain at a large-scale level. Also, the ITS community needs to focus on creating more benchmark datasets for different research tasks related to DL applications. Although PeMS has been used as a popular dataset for traffic state prediction as shown in 1, the absence of any comparable benchmark dataset

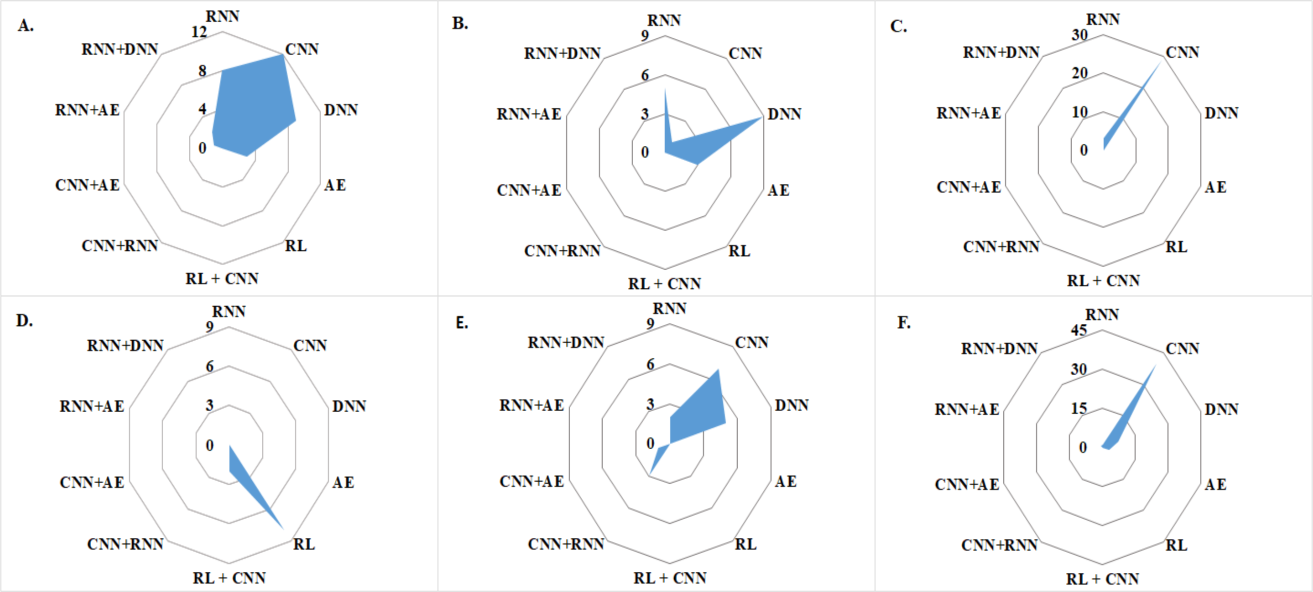


Figure 5: ITS vs DL models - A:Traffic Character, B:Traffic Incident, C:Vehicle ID, D:Traffic Signal, E:Public Transport, F:Visual Recognition

for traffic incident inference and ride-sharing studies has resulted in most of these studies using an original dataset. This has created difficulties in comparing different algorithms to determine the state-of-the-art model. Indeed, one of the reasons these research areas have still not been significantly explored using DL models is likely attributable to their lack of a recognized benchmark dataset. While this study has shown that DL models have been successfully applied to traffic state prediction, vehicle ID and visual recognition tasks, significant improvements need to be made in the use of DL models for other research topics such as traffic incident inference, traffic signal timing, ride sharing and other public transportation concerns. These topics have still not been fully explored using DL models and hence there remains significant scope for improving detection and prediction accuracy in these areas.

While DL models are becoming increasingly popular among researchers as the most effective classification method in visual recognition tasks in the ITS domain, privacy and security are extremely important. Therefore, the potential for adversarial attacks and thus the need for robustifying DL models have been receiving greater attention. (Adversarial attacks in this domain are, in most of the cases, small changes in the input which are imperceptible to the human eye but make the classifier classify incorrectly.) For example, self-driving cars use DL algorithms to recognize traffic signs [248] , other vehicles and related objects for navigation purposes. However, if DL models fail to detect a stop sign due to slight modification in a couple pixels, this can create serious impedance to the adoption of self-driving cars. Adversarial attacks are therefore an increasing area of focus in different DL application research topics such as natural language processing, computer vision, speech recognition and malware detection [249, 250, 251, 252, 253].

[254] has called into question the advisability of using neural networks and SVMs in security-sensitive applications, demonstrating the legitimacy of their concern by attacking some arbitrary PDF files and the MNIST dataset using the gradient descent evasion attack algorithm that they proposed. Their suggested solution is employing regularization terms in classifiers. In the same vein [255] has shown that accuracy for perturbed input due to adversarial attacks is much less than that in the case of high magnitude noise. Another downside of DL classification methods is that adversarial attacks can be independent of the classification model, meaning that one can generate an adversarial attack that can fool a machine learning system without any access to the model. These are called black-box attacks, a concept first introduced by [256], whereas white-box attacks are when the attacker is aware of all relevant information such as the training dataset, the model, etc. For example, [257] has used a Projected Gradient Descent (PGD) form of attack, which is different from related work that has mostly used a form of attack involving the Fast Gradient Sign Method (FGSM). Also, [258] has come up with a systematic way to compute universal attacks that are small image-agnostic perturbations that have a high probability of breaking most classifiers. Concurrent to research regarding designing attacks and understanding the vulnerability of neural networks to them, researchers have studied different ways to defend against adversarial attacks to make DNNs robust to them. One of the most popular approaches to defense against adversarial attacks is to add the adversarial set generated by any algorithm to the training set and then training the neural network with the new augmented dataset [259]. [260] has shown that although this method works for specific perturbations, networks being trained by this method are not robust to all adversaries. For example, while working to mitigate the effect of adversaries using denoising autoencoders (DAEs), [261] discovered that the resulting DNN became even more

sensitive to perturbed input data.

Around the same time, [262] designed a metric to measure the robustness of networks and approximate this using the encoding of their robustness as a linear program to improve the robustness of the overall DNN. Defense against adversarial attacks can be looked at as a robust optimization problem, as [263] has shown that adversarial training using their robust optimization algorithm results in the robustification of network optimization by increasing the accuracy and robustness of the DNN. Another recent method to harden DNNs against adversarial attacks is defensive distillation which has shown outstanding preliminary results in being able to reduce the adversarial attack success rate from 95% to 0.5% [264], but [265] defeated this method by designing a powerful attack able to break this defense mechanism. Thus, defense and design against adversarial attacks remain an open problem in DL applications.

As mentioned above, most studies regarding the application of DL models in transportation have paid no attention to robustness. However, in light of emerging malware attacks, the importance of defending models from such attacks has become increasingly important. These attacks usually destroy the input data by adding noise to them. These attacks can thus disturb the control unit by causing it to infer wrong information from the data, resulting in serious accidents. Also, another source of noise can be the weather conditions such as rainy or snowy conditions. Increasing the robustness of detection models will enable ITS models to operate better in severe conditions and thus improve their performance.

In summary, though much research is happening in various domains of ITS using a variety of DL models, the focus of future research in DL for ITS should encompass the following: how to develop DL models able to efficiently use the heterogeneous ITS data generated, how to build robust detection models and how to ensure security and privacy in the use of these models.

6. References

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, T. Tang, Big data analytics in intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems (2018) 1–16.

[2] R. P. Loce, E. A. Bernal, W. Wu, R. Bala, Computer vision in roadway transportation systems : A survey, Journal of Electronic Imaging 22 (2013) 041121.

[3] H. Nguyen, L.-M. Kieu, T. Wen, C. Cai, Deep learning methods in transportation domain: A review, IET Intelligent Transport Systems 12 (2018) 998–1004.

[4] Y. Wang, D. Zhang, Y. Liu, B. Dai, L. H. Lee, Enhancing transportation systems via deep learning: A survey, Transportation Research Part C: Emerging Technologies (2018).

[5] M. Dougherty, A review of neural networks applied to transport, Transportation Research Part C: Emerging Technologies 3 (1995) 247–260.

[6] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement, Annals of Internal Medicine 151 (2009) 264–269.

[7] Home - transport research international documentation, 2017. URL: https://trid.trb.org/.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252.

[9] N. Ketkar, et al., Deep Learning with Python, Springer, 2017.

[10] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[11] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks 61 (2015) 85–117.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation 1 (1989) 541–551.

[13] D. H. Hubel, T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, The Journal of Physiology 160 (1962) 106–154.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[15] Y. Bengio, I. J. Goodfellow, A. Courville, Deep learning, Nature 521 (2015) 436–444.

[16] R. Yamashita, M. Nishio, R. K. G. Do, K. Togashi, Convolutional neural networks: An overview and application in radiology, Insights into Imaging 9 (2018) 611–629.

[17] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Transactions on Medical Imaging 35 (2016) 1285–1298.

[18] Z. C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, arXiv preprint arXiv:1506.00019 (2015).

[19] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79 (1982) 2554–2558.

[20] J. L. Elman, Finding structure in time, Cognitive Science 14 (1990) 179–211.

[21] R. J. Williams, D. Zipser, Experimental analysis of the real-time recurrent learning algorithm, Connection Science 1 (1989) 87–111.

[22] S. Hochreiter, J. Schmidhuber, LSTM can solve hard long time lag problems, in: Advances in Neural Information Processing Systems, 1997, pp. 473–479.

[23] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks 18 (2005) 602–610.

[24] F. A. Gers, E. Schmidhuber, LSTM recurrent networks learn simple context-free and context-sensitive languages, IEEE Transactions on Neural Networks 12 (2001) 1333–1340.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[26] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1096–1103.

[27] C. Tan, C. Eswaran, Reconstruction of handwritten digit images using autoencoder neural networks, in: 2008 Canadian Conference on Electrical and Computer Engineering, IEEE, 2008, pp. 000465–000470.

[28] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[29] C. Doersch, Tutorial on variational autoencoders, arXiv preprint arXiv:1606.05908 (2016).

[30] Q. V. Le, et al., A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks, Google Brain (2015) 1–20.

[31] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 37–49.

[32] K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Processing Magazine 34 (2017) 26–38.

[33] D. Silver, UCL Course on RL , Lecture Notes: REINFORCEMENT LEARNING, 2015. URL: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html. Last visited on 2019/09/09.

[34] G. Tesauro, Temporal difference learning and td-gammon, Communications of the ACM 38 (1995) 58–68.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529.

[36] L.-J. Lin, Self-improving reactive agents based on reinforcement learning, planning and teaching, Machine Learning 8 (1992) 293–321.

[37] H. V. Hasselt, Double q-learning, in: Advances in Neural Information Processing Systems, 2010, pp. 2613–2621.

[38] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, N. De Freitas, Dueling network architectures for deep reinforcement learning, arXiv preprint arXiv:1511.06581 (2015).

[39] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.

[40] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.

[41] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160 (2016).

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[43] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434 (2015).

[44] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[45] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, IEEE Signal Processing Magazine 35 (2018) 53–65.

[46] J. S. Sussman, Perspectives on intelligent transportation systems (ITS), Springer Science & Business Media, 2008.

[47] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: A generic approach for extreme condition traffic forecasting, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 777–785.

[48] Y. Chen, L. Shu, L. Wang, Traffic flow prediction with big data: A deep learning based time series model, in: 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2017, pp. 1010–1011.

[49] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: Deep belief networks with multitask learning, IEEE Transactions on Intelligent Transportation Systems 15 (2014) 2191–2201.

[50] M. Khajeh Hosseini, A. Talebpour, Traffic prediction using time-space diagram: A convolutional neural network approach, Transportation Research Record (2019) 0361198119841291.

[51] C. Siripanpornchana, S. Panichpapiboon, P. Chaovalit, Travel-time prediction with deep learning, in: 2016 IEEE Region 10 Conference (TENCON), IEEE, 2016, pp. 1859–1862.

[52] Y. Hou, P. Edara, Network scale travel time prediction using deep learning, Transportation Research Record 2672 (2018) 115–123.

[53] A. Koesdwiady, R. Soua, F. Karray, Improving traffic flow prediction with weather information in connected cars: A deep learning approach, IEEE Transactions on Vehicular Technology 65 (2016) 9508–9517.

[54] D. Zhang, M. R. Kabuka, Combining weather condition data to predict traffic flow: A gru-based deep learning approach, IET Intelligent Transport Systems 12 (2018) 578–585.

[55] Y. Jia, J. Wu, Y. Du, Traffic speed prediction using deep learning method, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1217–1222.

[56] Y. Liu, Y. Wang, X. Yang, L. Zhang, Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 1–8.

[57] Y. Tian, L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network, in: 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity), IEEE, 2015, pp. 153–158.

[58] X. Ma, H. Yu, Y. Wang, Y. Wang, Large-scale transportation network congestion evolution prediction using deep learning theory, PloS one 10 (2015) e0119044.

[59] N. G. Polson, V. O. Sokolov, Deep learning for short-term traffic flow prediction, Transportation Research Part C: Emerging Technologies 79 (2017) 1–17.

[60] J. Wang, Q. Gu, J. Wu, G. Liu, Z. Xiong, Traffic speed prediction and congestion source exploration: A deep learning method, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 499–508.

[61] M. Fouladgar, M. Parchami, R. Elmasri, A. Ghaderi, Scalable deep traffic flow neural networks for urban traffic congestion prediction, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2251–2258.

[62] S. Du, T. Li, X. Gong, Y. Yang, S. J. Horng, Traffic flow forecasting based on hybrid deep learning framework, in: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, 2017, pp. 1–6.

[63] S. Goudarzi, M. Kama, M. Anisi, S. Soleymani, F. Doctor, Self-organizing traffic flow prediction with an optimized deep belief network for internet of vehicles, Sensors 18 (2018) 3459.

[64] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254 (2018).

[65] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (2017) 818.

[66] B. Yu, Z. Guo, S. Asian, H. Wang, G. Chen, Flight delay prediction for commercial air transport: A deep learning approach, Transportation Research Part E: Logistics and Transportation Review 125 (2019) 203–221.

[67] Z. Cui, R. Ke, Y. Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, arXiv preprint arXiv:1801.02143 (2018).

[68] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: A deep learning approach, IEEE Transactions on Intelligent Transportation Systems 16 (2014) 865–873.

[69] Y. Duan, Y. Lv, Y.-L. Liu, F.-Y. Wang, An efficient realization of deep learning for traffic data imputation, Transportation Research Part C: Emerging Technologies 72 (2016) 168–181.

[70] X. Gang, W. Kang, F. Wang, F. Zhu, Y. Lv, X. Dong, J. Riekki, S. Pirttikangas, Continuous travel time prediction for transit signal priority based on a deep network, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, IEEE, 2015, pp. 523–528.

[71] F. Ding, Z. Zhang, Y. Zhou, X. Chen, B. Ran, Large-scale full-coverage traffic speed estimation under extreme traffic conditions using a big data and deep learning approach: Case study in china, Journal of Transportation Engineering, Part A: Systems 145 (2019) 05019001.

[72] J. Wang, R. Chen, Z. He, Traffic speed prediction for urban transportation network: A path based deep learning approach, Transportation Research Part C: Emerging Technologies 100 (2019) 372–385.

[73] H. Nguyen, C. Bentley, L. M. Kieu, Y. Fu, C. Cai, Deep learning system for travel speed predictions on multiple arterial road segments, Transportation Research Record (2019) 0361198119838508.

[74] D. Jo, B. Yu, H. Jeon, K. Sohn, Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies, IEEE Transactions on Vehicular Technology 68 (2018) 1188–1197.

[75] Y. Gu, W. Lu, L. Qin, M. Li, Z. Shao, Short-term prediction of lane-level traffic speeds: A fusion deep learning model, Transportation research part C: emerging technologies 106 (2019) 1–16.

[76] Z. Zhang, M. Li, X. Lin, Y. Wang, F. He, Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies, Transportation Research Part C: Emerging Technologies 105 (2019) 297–322.

[77] Y. Jia, J. Wu, M. Xu, Traffic flow prediction with rainfall impact using a deep learning method, Journal of Advanced Transportation 2017 (2017).

[78] Z. Zhao, W. Chen, X. Wu, P. C. Chen, J. Liu, LSTM network: A deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2017) 68–75.

[79] G. Yang, Y. Wang, H. Yu, Y. Ren, J. Xie, Short-term traffic state prediction based on the spatiotemporal features of critical road sections, Sensors 18 (2018) 2287.

[80] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, Transportation Research Part C: Emerging Technologies 90 (2018) 166–180.

[81] Y. Zhuang, R. Ke, Y. Wang, Innovative method for traffic data imputation based on convolutional neural network, IET Intelligent Transport Systems (2018).

[82] M. Arif, G. Wang, S. Chen, Deep learning with non-parametric regression model for traffic flow prediction, in: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, 2018, pp. 681–688.

[83] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, arXiv preprint arXiv:1709.04875 (2017).

[84] Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, arXiv preprint arXiv:1802.07007 (2018).

[85] S. Lee, K. Xie, D. Ngoduy, M. Keyvan-Ekbatani, H. Yang, A lane-based predictive model of downstream arrival rates in a queue estimation model using a long short-term memory network, Transportation Research Procedia 34 (2018) 163–170.

[86] X. Dai, R. Fu, E. Zhao, Z. Zhang, Y. Lin, . F.-Y. Wang, L. Li, Deeptrend 2.0: A light-weighted multi-scale traffic prediction model using detrending, Transportation Research Part C: Emerging Technologies 103 (2019) 142–157.

[87] M. H. Tawfeek, K. El-Basyouny, Estimating traffic volume on minor roads at rural stop-controlled intersections using deep learning, Transportation Research Record (2019) 0361198119837236.

[88] Y. Lin, X. Dai, L. Li, F.-Y. Wang, Pattern sensitive prediction of traffic flow based on generative adversarial framework, IEEE Transactions on Intelligent Transportation Systems 20 (2018) 2395–2400.

[89] J. Li, J. Wang, Short term traffic flow prediction based on deep learning, in: CICTP 2019, 2017, pp. 2457–2469.

[90] X. Zhao, Y. Gu, L. Chen, Z. Shao, Urban short-term traffic flow prediction based on stacked autoencoder, in: 19th COTA International Conference of Transportation Professionals, 2019.

[91] M. Chen, X. Yu, Y. Liu, Pcnn: Deep convolutional networks for short-term traffic congestion prediction, IEEE Transactions on Intelligent Transportation Systems (2018) 1–10.

[92] T. Pamula, Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks, IEEE Intelligent Transportation Systems Magazine 10 (2018) 11–21.

[93] P. Chakraborty, Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, S. Sarkar, Traffic congestion detection from camera images using deep convolution neural networks, Transportation Research Record 2672 (2018) 222–231.

[94] J. Guo, Y. Liu, Y. Wang, K. Yang, Deep learning based congestion prediction using probe trajectory data, in: 19th COTA International Conference of Transportation Professionals, 2019.

[95] J. Chung, K. Sohn, Image-based learning to measure traffic density using a deep convolutional neural network, IEEE Transactions on Intelligent Transportation Systems 19 (2017) 1670–1675.

[96] Q. Chen, X. Song, H. Yamada, R. Shibasaki, Learning deep representation from big and heterogeneous data for traffic accident inference, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[97] H. Ren, Y. Song, J. Liu, Y. Hu, J. Lei, A deep learning approach to the prediction of short-term traffic accident risk, arXiv preprint arXiv:1710.09543 (2017).

[98] J. Bao, P. Liu, S. V. Ukkusuri, A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data, Accident Analysis & Prevention 122 (2019) 239–254.

[99] E. M. Ali, M. M. Ahmed, S. S. Wulff, Detection of critical safety events on freeways in clear and rainy weather using shrp2 naturalistic driving data: Parametric and non-parametric techniques, Safety Science (2019).

[100] X. Ni, H. Wang, C. Che, J. Hong, Z. Sun, Civil aviation safety evaluation based on deep belief network and principal component analysis, Safety Science 112 (2019) 90–95.

[101] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, R. Mantilla, Predicting traffic accidents through heterogeneous urban data: A case study, in: Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017), Halifax, NS, Canada, volume 14, 2017.

[102] G. Pan, L. Fu, L. Thakali, Development of a global road safety performance function using deep neural networks, International Journal of Transportation Science and Technology 6 (2017) 159–173.

[103] C. Dong, C. Shao, J. Li, Z. Xiong, An improved deep learning model for traffic crash prediction, Journal of Advanced Transportation 2018 (2018).

[104] Z. Yuan, X. Zhou, T. Yang, Hetero-convLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 984–992.

[105] C. Chen, H. Xiang, T. Qiu, C. Wang, Y. Zhou, V. Chang, A rear-end collision prediction scheme based on deep learning in the internet of vehicles, Journal of Parallel and Distributed Computing 117 (2018) 192–204.

[106] G. Pan, L. Fu, L. Thakali, M. Muresan, M. Yu, An improved deep belief network model for road safety analyses, arXiv preprint arXiv:1812.07410 (2018).

[107] T. Suzuki, H. Kataoka, Y. Aoki, Y. Satoh, Anticipating traffic accidents with adaptive loss and large-scale incident db, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3521–3529.

[108] A. Theofilatos, C. Chen, C. Antoniou, Comparing machine learning and deep learning methods for real-time crash prediction, Transportation Research Record (2019) 0361198119841571.

[109] C. El Hatri, J. Boumhidi, Fuzzy deep learning based urban traffic incident detection, Cognitive Systems Research 50 (2018) 206–213.

[110] Z. Zhang, Q. He, J. Gao, M. Ni, A deep learning approach for detecting traffic accidents from social media data, Transportation Research Part C: Emerging Technologies 86 (2018) 580–596.

[111] M. Sameen, B. Pradhan, Severity prediction of traffic accidents with recurrent neural networks, Applied Sciences 7 (2017) 476.

[112] S. Alkheder, M. Taamneh, S. Taamneh, Severity prediction of traffic accident using an artificial neural network, Journal of Forecasting 36 (2017) 100–108.

[113] S. Das, A. Dutta, K. Dixon, L. Minjares-Kyle, G. Gillette, Using deep learning in severity analysis of at-fault motorcycle rider crashes, Transportation Research Record 2672 (2018) 122–134.

[114] D. Singh, C. K. Mohan, Deep spatio-temporal representation for detection of road accidents using stacked autoencoder, IEEE Transactions on Intelligent Transportation Systems (2018) 1–9.

[115] D. Zang, Z. Chai, J. Zhang, D. Zhang, J. Cheng, Vehicle license plate recognition using visual attention model and deep learning, Journal of Electronic Imaging 24 (2015) 033001.

[116] M. Z. Abedin, A. C. Nath, P. Dhar, K. Deb, M. S. Hossain, License plate recognition system based on contour properties and deep learning model, in: 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), IEEE, 2017, pp. 590–593.

[117] X. Liu, W. Liu, T. Mei, H. Ma, A deep learning-based approach to progressive vehicle re-identification for urban surveillance, in: European Conference on Computer Vision, Springer, 2016, pp. 869–884.

[118] C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, L. Mei, Vehicle color recognition with spatial pyramid deep learning, IEEE Transactions on Intelligent Transportation Systems 16 (2015) 2925–2934.

[119] Z. Tang, G. Wang, H. Xiao, A. Zheng, J.-N. Hwang, Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 108–115.

[120] Y. Zheng, K. Ozcan, S. Velipasalar, A codebook of brightness transfer functions for improved target re-identification across non-overlapping camera views, in: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2017, pp. 166–170.

[121] S. Yu, Y. Wu, W. Li, Z. Song, W. Zeng, A model for fine-grained vehicle classification based on deep learning, Neurocomputing 257 (2017) 97–103.

[122] S. Arabi, A. Haghighat, A. Sharma, A deep-learning-based computer vision solution for construction vehicle detection, Computer-Aided Civil and Infrastructure Engineering 35 (2020) 753–767.

[123] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, S.-Y. Chien, Vehicle re-identification with the space-time prior, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 121–128.

[124] O. Bulan, V. Kozitsky, P. Ramesh, M. Shreve, Segmentation-and annotation-free license plate recognition with deep localization and failure identification, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 2351–2363.

[125] Y. Yang, D. Li, Z. Duan, Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features, IET Intelligent Transport Systems 12 (2017) 213–219.

[126] W. Puarungroj, N. Boonsirisumpun, Thai license plate recognition based on deep learning, Procedia Computer Science 135 (2018) 214–221.

[127] L. Xie, T. Ahmad, L. Jin, Y. Liu, S. Zhang, A new CNN-based method for multi-directional car license plate detection, IEEE Transactions on Intelligent Transportation Systems 19 (2018) 507–517.

[128] Y. Huang, R. Wu, Y. Sun, W. Wang, X. Ding, Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy, IEEE Transactions on Intelligent Transportation Systems 16 (2015) 1951–1960.

[129] Z. Dong, Y. Wu, M. Pei, Y. Jia, Vehicle type classification using a semisupervised convolutional neural network, IEEE Transactions on Intelligent Transportation Systems 16 (2015) 2247–2256.

[130] J. Zhong, T. Lei, G. Yao, Robust vehicle detection in aerial images based on cascaded convolutional neural networks, Sensors 17 (2017) 2720.

[131] J. Fang, Y. Zhou, Y. Yu, S. Du, Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture, IEEE Transactions on Intelligent Transportation Systems 18 (2016) 1782–1792.

[132] Y. O. Adu-Gyamfi, S. K. Asare, A. Sharma, T. Titus, Automated vehicle recognition with deep convolutional neural networks, Transportation Research Record 2645 (2017) 113–122.

[133] Q. Hu, H. Wang, T. Li, C. Shen, Deep CNNs with spatially weighted pooling for fine-grained car recognition, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 3147–3156.

[134] F. C. Soon, H. Y. Khaw, J. H. Chuah, J. Kanesan, Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition, IET Intelligent Transport Systems 12 (2018) 939–946.

[135] Y. Li, B. Song, X. Kang, X. Du, M. Guizani, Vehicle-type detection based on compressed sensing and deep learning in vehicular networks, Sensors 18 (2018) 4500.

[136] Y. Wang, C. Wang, H. Zhang, Ship classification in high-resolution sar images using deep learning of small datasets, Sensors 18 (2018) 2929.

[137] J. Wang, H. Zheng, Y. Huang, X. Ding, Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning, IEEE Transactions on Intelligent Transportation Systems (2017) 1–10.

[138] X. Xiang, N. Lv, X. Guo, S. Wang, A. El Saddik, Engineering vehicles detection based on modified faster R-CNN for power grid surveillance, Sensors 18 (2018) 2258.

[139] M. Simoncini, L. Taccari, F. Sambo, L. Bravi, S. Salti, A. Lori, Vehicle classification from low-frequency gps data with recurrent neural networks, Transportation Research Part C: Emerging Technologies 91 (2018) 176–191.

[140] Z. Yan, Y. Feng, C. Cheng, J. Fu, X. Zhou, J. Yuan, Extensive exploration of comprehensive vehicle attributes using D-CNN with weighted multi-attribute strategy, IET Intelligent Transport Systems 12 (2017) 186–193.

[141] L. Suhao, L. Jinzhao, L. Guoquan, B. Tong, W. Huiqian, P. Yu, Vehicle type detection based on deep learning in traffic scene, Procedia Computer Science 131 (2018) 564–572.

[142] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Vision-based occlusion handling and vehicle classification for traffic surveillance systems, IEEE Intelligent Transportation Systems Magazine 10 (2018) 80–92.

[143] K. F. Hussain, M. Afifi, G. Moussa, A comprehensive study of the effect of spatial resolution and color of digital images on vehicle classification, IEEE Transactions on Intelligent Transportation Systems (2018) 1–10.

[144] F. C. Soon, H. Y. Khaw, J. H. Chuah, J. Kanesan, Pcanet-based convolutional neural network architecture for a vehicle model recognition system, IEEE Transactions on Intelligent Transportation Systems (2018) 1–11.

[145] Y. Shen, T. Xiao, H. Li, S. Yi, X. Wang, Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1900–1909.

[146] R. Tapu, B. Mocanu, T. Zaharia, Deep-see: Joint object detection, tracking and recognition with application to visually impaired navigational assistance, Sensors 17 (2017) 2473.

[147] P. A. Marín-Reyes, L. Bergamini, J. Lorenzo-Navarro, A. Palazzi, S. Calderara, R. Cucchiara, Unsupervised vehicle re-identification using triplet networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2018, pp. 166–1665.

[148] S. Choi, H. Yeo, J. Kim, Network-wide vehicle trajectory prediction in urban traffic networks using deep learning, Transportation Research Record 2672 (2018) 173–184.

[149] E.-J. Kim, H.-C. Park, S.-W. Ham, S.-Y. Kho, D.-K. Kim, Extracting vehicle trajectories using unmanned aerial vehicles in congested traffic conditions, Journal of Advanced Transportation 2019 (2019).

[150] L. Li, Y. Lv, F.-Y. Wang, Traffic signal timing via deep reinforcement learning, IEEE/CAA Journal of Automatica Sinica 3 (2016) 247–254.

[151] J. Gao, Y. Shen, J. Liu, M. Ito, N. Shiratori, Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network, arXiv preprint arXiv:1705.02755 (2017).

[152] X. Liang, X. Du, G. Wang, Z. Han, Deep reinforcement learning for traffic light control in vehicular networks, arXiv preprint arXiv:1803.11115 (2018).

[153] W. Genders, S. Razavi, Evaluating reinforcement learning state representations for adaptive traffic signal control, Procedia Computer Science 130 (2018) 26–33.

[154] H. Wei, G. Zheng, H. Yao, Z. Li, Intellilight: A reinforcement learning approach for intelligent traffic light control, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2496–2505.

[155] E. Van der Pol, F. A. Oliehoek, Coordinated deep reinforcement learners for traffic light control, Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016) (2016).

[156] S. S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intelligent Transport Systems 11 (2017) 417–423.

[157] C.-H. Wan, M.-C. Hwang, Value-based deep reinforcement learning for adaptive isolated intersection signal control, IET Intelligent Transport Systems 12 (2018) 1005–1010.

[158] M. Muresan, L. Fu, G. Pan, Adaptive traffic signal control with deep reinforcement learning an exploratory investigation, arXiv preprint arXiv:1901.00960 (2019).

[159] X. Liang, X. Du, G. Wang, Z. Han, A deep reinforcement learning network for traffic light cycle control, IEEE Transactions on Vehicular Technology 68 (2019) 1243–1253.

[160] Y. Gong, M. Abdel-Aty, Q. Cai, M. S. Rahman, Decentralized network level adaptive signal control by multi-agent deep reinforcement learning, Transportation Research Interdisciplinary Perspectives 1 (2019) 100020.

[161] R. Huang, J. Hu, Y. Huo, X. Pei, Cooperative multi-intersection traffic signal control based on deep reinforcement learning, in: CICTP 2019, ????, pp. 2959–2970.

[162] I. Saadi, M. Wong, B. Farooq, J. Teller, M. Cools, An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service, arXiv preprint arXiv:1703.02433 (2017).

[163] A. Dominguez-Sanchez, M. Cazorla, S. Orts-Escolano, Pedestrian movement direction recognition using convolutional neural networks, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 3540–3548.

[164] J. Jung, K. Sohn, Deep-learning architecture to forecast destinations of bus passengers from entry-only smart-card data, IET Intelligent Transport Systems 11 (2017) 334–339.

[165] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, H. He, Robot-assisted pedestrian regulation based on deep reinforcement learning, IEEE Transactions on Cybernetics (2018).

[166] H. Zhu, X. Yang, Y. Wang, Prediction of daily entrance and exit passenger flow of rail transit stations by deep learning method, Journal of Advanced Transportation 2018 (2018).

[167] L. Liu, R.-C. Chen, A novel passenger flow prediction model using deep learning methods, Transportation Research Part C: Emerging Technologies 84 (2017) 74–91.

[168] J. Ke, H. Zheng, H. Yang, X. M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, Transportation Research Part C: Emerging Technologies 85 (2017) 591–608.

[169] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, IEEE, 2017.

[170] S. Liao, L. Zhou, X. Di, B. Yuan, J. Xiong, Large-scale short-term urban taxi demand forecasting using deep learning, in: Proceedings of the 23rd Asia and South Pacific Design Automation Conference, IEEE Press, 2018, pp. 428–433.

[171] L. Lin, Z. He, S. Peeta, Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach, Transportation Research Part C: Emerging Technologies 97 (2018) 258–276.

[172] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[173] X. Ma, J. Zhang, B. Du, C. Ding, L. Sun, Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction, IEEE Transactions on Intelligent Transportation Systems 20 (2018) 2278–2288.

[174] D. Nam, H. Kim, J. Cho, R. Jayakrishnan, A model based on deep learning for predicting travel mode choice, in: Proceedings of the Transportation Research Board 96th Annual Meeting Transportation Research Board, Washington, DC, USA, 2017, pp. 8–12.

[175] D. Shi, J. Ding, S. M. Errapotu, H. Yue, W. Xu, X. Zhou, M. Pan, Deep q-network based route scheduling for transportation network company vehicles, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 1–7.

[176] S. Dabiri, K. Heaslip, Inferring transportation modes from gps trajectories using a convolutional neural network, Transportation Research Part C: Emerging Technologies 86 (2018) 360–371.

[177] Y. Cui, C. Meng, Q. He, J. Gao, Forecasting current and next trip purpose with social media data and google places, Transportation Research Part C: Emerging Technologies 97 (2018) 159–174.

[178] Y. Liu, Z. Liu, R. Jia, Deeppf: A deep learning based architecture for metro passenger flow prediction, Transportation Research Part C: Emerging Technologies 101 (2019) 18–34.

[179] S. Hao, D.-H. Lee, D. Zhao, Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system, Transportation Research Part C: Emerging Technologies 107 (2019) 287–300.

[180] C. Xu, J. Ji, P. Liu, The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets, Transportation Research Part C: Emerging Technologies 95 (2018) 47–60.

[181] L. Zhang, G. Zhang, Z. Liang, E. F. Ozioko, Multi-features taxi destination prediction with frequency domain processing, PloS one 13 (2018) e0194629.

[182] J. Wen, J. Zhao, P. Jaillet, Rebalancing shared mobility-on-demand systems: A reinforcement learning approach, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 220–225.

[183] H. Yang, K. Xie, K. Ozbay, Y. Ma, Z. Wang, Use of deep learning to predict daily usage of bike sharing systems, Transportation Research Record (2018) 0361198118801354.

[184] T. S. Kim, W. K. Lee, S. Y. Sohn, Graph convolutional network approach applied to predict hourly bike-sharing demands considering spatial, temporal, and global effects, PloS one 14 (2019) e0220782.

[185] T. Kim, J. Ghosh, Robust detection of non-motorized road users using deep learning on optical and lidar data, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 271–276.

[186] A. Dairi, F. Harrou, Y. Sun, M. Senouci, Obstacle detection for intelligent transportation systems using deep stacked autoencoder and *k*-nearest neighbor scheme, IEEE Sensors Journal 18 (2018) 5122–5132.

[187] A. Dairi, F. Harrou, M. Senouci, Y. Sun, Unsupervised obstacle detection in driving environments using deep-learning-based stereovision, Robotics and Autonomous Systems 100 (2018) 287–301.

[188] H. Wang, Y. Cai, X. Chen, L. Chen, Night-time vehicle sensing in far infrared image with deep learning, Journal of Sensors 2016 (2016).

[189] Y. Cai, X. Sun, H. Wang, L. Chen, H. Jiang, Night-time vehicle detection algorithm based on visual saliency and deep learning, Journal of Sensors 2016 (2016).

[190] J.-G. Wang, L. Zhou, Y. Pan, S. Lee, Z. Song, B. S. Han, V. B. Saputra, Appearance-based brake-lights recognition using deep learning and vehicle detection, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2016, pp. 815–820.

[191] R. Qian, B. Zhang, Y. Yue, Z. Wang, F. Coenen, Robust chinese traffic sign detection and recognition with deep convolutional neural network, in: 2015 11th International Conference on Natural Computation (ICNC), IEEE, 2015, pp. 791–796.

[192] Y. Yang, H. Luo, H. Xu, F. Wu, Towards real-time traffic sign detection and classification, IEEE Transactions on Intelligent transportation systems 17 (2015) 2022–2031.

[193] F. Lin, Y. Lai, L. Lin, Y. Yuan, A traffic sign recognition method based on deep visual feature, in: 2016 Progress in Electromagnetic Research Symposium (PIERS), IEEE, 2016, pp. 2247–2250.

[194] K. Lim, Y. Hong, Y. Choi, H. Byun, Real-time traffic sign recognition based on a general purpose gpu and deep-learning, PLoS one 12 (2017) e0173317.

[195] Y. Zeng, X. Xu, D. Shen, Y. Fang, Z. Xiao, Traffic sign recognition using kernel extreme learning machines with deep perceptual features, IEEE Transactions on Intelligent Transportation Systems 18 (2016) 1647–1653.

[196] W. Hu, Q. Zhuo, C. Zhang, J. Li, Fast branch convolutional neural network for traffic sign recognition, IEEE Intelligent Transportation Systems Magazine 9 (2017) 114–126.

[197] Y. Yuan, Z. Xiong, Q. Wang, An incremental framework for video-based traffic sign detection, tracking, and recognition, IEEE Transactions on Intelligent Transportation Systems 18 (2016) 1918–1929.

[198] Á. Arcos-García, J. A. Álvarez-García, L. M. Soria-Morillo, Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods, Neural Networks 99 (2018) 158–165.

[199] S. Natarajan, A. K. Annamraju, C. S. Baradkar, Traffic sign recognition using weighted multi-convolutional neural network, IET Intelligent Transport Systems 12 (2018) 1396–1405.

[200] H. S. Lee, K. Kim, Simultaneous traffic sign detection and boundary estimation using convolutional neural network, IEEE Transactions on Intelligent Transportation Systems 19 (2018) 1652–1663.

[201] C. Li, Z. Chen, Q. J. Wu, C. Liu, Deep saliency with channel-wise hierarchical feature responses for traffic sign detection, IEEE Transactions on Intelligent Transportation Systems (2018).

[202] C. You, C. Wen, C. Wang, J. Li, A. Habib, Joint 2-d-3-d traffic sign landmark data set for geo-localization using mobile laser scanning data, IEEE Transactions on Intelligent Transportation Systems (2018).

[203] Z. Lin, M. Yih, J. M. Ota, J. Owens, P. Muyan-Ozcelik, Benchmarking deep learning frameworks and investigating fpga deployment for traffic sign classification and detection, IEEE Transactions on Intelligent Vehicles (2019).

[204] C. Li, C. Yang, The research on traffic sign recognition based on deep learning, in: 2016 16th International Symposium on Communications and Information Technologies (ISCIT), IEEE, 2016, pp. 156–161.

[205] M. Weber, P. Wolf, J. M. Zöllner, Deeptlr: A single deep convolutional network for detection and classification of traffic lights, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2016, pp. 342–348.

[206] K. Behrendt, L. Novak, R. Botros, A deep learning approach to traffic lights: Detection, tracking, and classification, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 1370–1377.

[207] H.-K. Kim, J. H. Park, H.-Y. Jung, An efficient color space for deep-learning based traffic light recognition, Journal of Advanced Transportation 2018 (2018).

[208] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, V. N. Murali, Deeplanes: End-to-end lane position estimation using deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 38–45.

[209] S. Arabi, B. Shafei, B. M. Phares, Fatigue analysis of sign-support structures during transportation under road-induced excitations, Engineering Structures 164 (2018) 305–315.

[210] S. Arabi, B. Shafei, B. M. Phares, Investigation of fatigue in steel sign-support structures under diurnal temperature changes, Journal of Constructional Steel Research 153 (2019) 286–297.

[211] G. Constantinescu, A. Bhatti, B. Phares, Effect of Wind Induced Unsteady Vortex Shedding, Diurnal Temperature Changes, and Transit Conditions on Truss Structures Supporting Large Highway Signs Problem Statement, Technical Report, 2018.

[212] S. Arabi, B. Shafei, B. M. Phares, Vulnerability assessment of sign support structures during transportation, Technical Report, 2017.

[213] J. Moll, P. Arnold, M. Mälzer, V. Krozer, D. Pozdniakov, R. Salman, S. Rediske, M. Scholz, H. Friedmann, A. Nuber, Radar-based structural health monitoring of wind turbine blades: The case of damage detection, Structural Health Monitoring 17 (2018) 815–822.

[214] K. Zhang, H. Cheng, B. Zhang, Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning, Journal of Computing in Civil Engineering 32 (2018) 04018001.

[215] S. Bang, S. Park, H. Kim, H. Kim, Encoder–decoder network for pixel-level road crack detection in black-box images, Computer-Aided Civil and Infrastructure Engineering (2019).

[216] A. Zhang, K. C. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li, B. Li, Deep learning–based fully automated pavement crack detection on 3d asphalt surfaces with an improved cracknet, Journal of Computing in Civil Engineering 32 (2018) 04018041.

[217] H. Nhat-Duc, Q.-L. Nguyen, V.-D. Tran, Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network, Automation in Construction 94 (2018) 203–213.

[218] V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, J. W. Jeon, Learning framework for robust obstacle detection, recognition, and tracking, IEEE Transactions on Intelligent Transportation Systems 18 (2016) 1633–1646.

[219] M. He, H. Luo, Z. Chang, B. Hui, Pedestrian detection with semantic regions of interest, Sensors 17 (2017) 2699.

[220] X. Li, Y. Liu, Z. Zhao, Y. Zhang, L. He, A deep learning approach of vehicle multitarget detection from traffic video, Journal of Advanced Transportation 2018 (2018).

[221] W. Zhang, Z. Wang, X. Liu, H. Sun, J. Zhou, Y. Liu, W. Gong, Deep learning-based real-time fine-grained pedestrian recognition using stream processing, IET Intelligent Transport Systems 12 (2018) 602–609.

[222] D. de Oliveira, M. Wehrmeister, Using deep learning and low-cost rgb and thermal cameras to detect pedestrians in aerial images captured by multirotor uav, Sensors 18 (2018) 2244.

[223] T. Ye, B. Wang, P. Song, J. Li, Automatic railway traffic object detection system using feature fusion refine neural network under shunting mode, Sensors 18 (2018) 1916.

[224] X. Zhang, L. Cheng, B. Li, H.-M. Hu, Too far to see? not really!—pedestrian detection with scale-aware localization policy, IEEE Transactions on Image Processing 27 (2018) 3703–3715.

[225] F. Zhou, J. Li, X. Li, Z. Li, Y. Cao, Freight car target detection in a complex background based on convolutional neural networks, Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit 233 (2019) 298–311.

[226] M. Rahman, M. Islam, J. Calhoun, M. Chowdhury, Real-time pedestrian detection approach with an efficient data communication bandwidth strategy, Transportation Research Record (2019) 0361198119843255.

[227] Y. Yao, B. Tian, F.-Y. Wang, Coupled multivehicle detection and classification with prior objectness measure, IEEE Transactions on Vehicular Technology 66 (2016) 1975–1984.

[228] T. Tang, S. Zhou, Z. Deng, H. Zou, L. Lei, Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining, Sensors 17 (2017) 336.

[229] X. Luo, R. Shen, J. Hu, J. Deng, L. Hu, Q. Guan, A deep convolution neural network model for vehicle recognition and face recognition, Procedia Computer Science 107 (2017) 715–720.

[230] Y. Zhou, L. Liu, L. Shao, M. Mellor, Fast automatic vehicle annotation for urban traffic surveillance, IEEE Transactions on Intelligent Transportation Systems 19 (2018) 1973–1984.

[231] H. Wang, Y. Yu, Y. Cai, L. Chen, X. Chen, A vehicle recognition algorithm based on deep transfer learning with a multiple feature subspace distribution, Sensors 18 (2018) 4109.

[232] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, P.-A. Heng, Sinet: A scale-insensitive convolutional neural network for fast vehicle detection, IEEE Transactions on Intelligent Transportation Systems 20 (2018) 1010–1019.

[233] R. V. Nezafat, O. Sahin, M. Cetin, Transfer learning using deep neural networks for classification of truck body types based on side-fire lidar data, Journal of Big Data Analytics in Transportation 1 (2019) 71–82.

[234] A. Shustanov, P. Yakimov, CNN design for real-time traffic sign recognition, Procedia Engineering 201 (2017) 718–725.

[235] Y. Y. Ye, X. L. Hao, H. J. Chen, Lane detection method based on lane structural analysis and CNNs, IET Intelligent Transport Systems 12 (2018) 513–520.

[236] Y. Dou, Y. Fang, C. Hu, R. Zheng, F. Yan, Gated branch neural network for mandatory lane changing suggestion at the on-ramps of highway, IET Intelligent Transport Systems 13 (2018) 48–54.

[237] X. Zhang, W. Yang, X. Tang, Y. Wang, Lateral distance detection model based on convolutional neural network, IET Intelligent Transport Systems 13 (2018) 31–39.

[238] L. Chen, X. Hu, T. Xu, H. Kuang, Q. Li, Turn signal detection during nighttime by CNN detector and perceptual hashing tracking, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 3303–3314.

[239] J. Park, K. Min, H. Kim, W. Lee, G. Cho, K. Huh, Road surface classification using a deep ensemble network with sensor feature selection, Sensors 18 (2018) 4342.

[240] T. M. Hoang, P. H. Nguyen, N. Q. Truong, Y. W. Lee, K. R. Park, Deep retinanet-based detection and classification of road markings by visible light camera sensors, Sensors 19 (2019) 281.

[241] Q. Wang, J. Gao, Y. Yuan, A joint convolutional neural networks and context transfer for street scenes labeling, IEEE Transactions on Intelligent Transportation Systems 19 (2017) 1457–1470.

[242] L. Li, B. Qian, J. Lian, W. Zheng, Y. Zhou, Traffic scene segmentation based on rgb-d image and deep learning, IEEE Transactions on Intelligent Transportation Systems 19 (2017) 1664–1669.

[243] B. Chen, C. Gong, J. Yang, Importance-aware semantic segmentation for autonomous vehicles, IEEE Transactions on Intelligent Transportation Systems (2018) 1–12.

[244] Y. Cai, D. Li, X. Zhou, X. Mou, Robust drivable road region detection for fixed-route autonomous vehicles using map-fusion images, Sensors 18 (2018) 4158.

[245] K. Wang, A. Zhang, J. Q. Li, Y. Fei, C. Chen, B. Li, Deep learning for asphalt pavement cracking recognition using convolutional neural network, in: Proc. Int. Conf. Airfield Highway Pavements, 2017, pp. 166–177.

[246] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, H.-J. Jung, Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle, Sensors 18 (2018) 1881.

[247] F. Daneshgaran, L. Zacheo, F. D. Stasio, M. Mondin, Use of deep learning for automatic detection of cracks in tunnels: Prototype-2 developed in the 2017–2018 time period, Transportation Research Record (2019) 0361198119845656.

[248] D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, arXiv preprint arXiv:1202.2745 (2012).

[249] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, Journal of Big Data 2 (2015) 1.

[250] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.

[251] Y. LeCun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, in: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, IEEE, 2010, pp. 253–256.

[252] L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: An overview, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 8599–8603.

[253] W. Hardy, L. Chen, S. Hou, Y. Ye, X. Li, Dl4md: A deep learning framework for intelligent malware detection, in: Proceedings of the International Conference on Data Mining (DMIN), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016, p. 61.

[254] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 387–402.

[255] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[256] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, A. Swami, Practical black-box attacks against deep learning systems using adversarial examples, arXiv preprint arXiv:1602.02697 1 (2016) 3.

[257] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).

[258] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1765–1773.

[259] T. Fawcett, In vivo spam filtering: A challenge problem for kdd, ACM SIGKDD Explorations Newsletter 5 (2003) 140–148.

[260] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).

[261] S. Gu, L. Rigazio, Towards deep neural network architectures robust to adversarial examples, arXiv preprint arXiv:1412.5068 (2014).

[262] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, A. Criminisi, Measuring neural net robustness with constraints, in: Advances in Neural Information Processing Systems, 2016, pp. 2613–2621.

[263] U. Shaham, Y. Yamada, S. Negahban, Understanding adversarial training: Increasing local stability of supervised models through robust optimization, Neurocomputing 307 (2018) 195–204.

[264] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 582–597.

[265] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57.