Transportation Research Part C 99 (2019) 144–163

Contents lists available at ScienceDirect

Transportation Research Part C

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

Review

Enhancing transportation systems via deep learning: A survey

Yuan Wang[a], Dongxiang Zhang[b,*], Ying Liu[b], Bo Dai[b], Loo Hay Lee[a]

[a] *Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore*
[b] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*

ARTICLE INFO

Keywords:
Deep learning
Transportation systems
Survey

ABSTRACT

Machine learning (ML) plays the core function to intellectualize the transportation systems. Recent years have witnessed the advent and prevalence of deep learning which has provoked a storm in ITS (Intelligent Transportation Systems). Consequently, traditional ML models in many applications have been replaced by the new learning techniques and the landscape of ITS is being reshaped. Under such perspective, we provide a comprehensive survey that focuses on the utilization of deep learning models to enhance the intelligence level of transportation systems. By organizing multiple dozens of relevant works that were originally scattered here and there, this survey attempts to provide a clear picture of how various deep learning models have been applied in multiple transportation applications.

1. Introduction

The rapid urbanization in the civil world has resulted in the sharp growth of population and vehicles in a city and imposed an ever-increasing burden on the transportation infrastructures. Consequently, traffic congestion has become a substantial threat to urban cities in terms of tremendous lost time and productivity, air pollution and wasted energy. Nowadays, it gradually forms a common sense that these problems can be solved, or at least alleviated, via new data technology. A tremendous number of sensors have been deployed, continuously generating streaming data that is required to be processed instantly to support real-time decision. There is hence an urgent demand to upgrade the existing transportation systems to a more advanced and intelligent level. Intelligent Transportation System (ITS) can be seen as an integrated transportation management system composed of advanced data communication, information processing and traffic management technology. It can instantly handle the real-time data collected from heterogeneous sources and analyze them for better decision making. For instance, IBM InfoSphere (Biem et al., 2010) can process 120,000 GPS points per second.

Machine learning techniques normally act as the function of brain in ITS and its accuracy and reliability straightly determine how intelligent the system is. In recent years, deep learning has witnessed an overwhelming success in computer vision, speed recognition and natural language processing. They have frequently broken new records of accuracy in a great number of applications. It is then a natural practice to apply the deep learning models as the classifier or predictor in ITS to enhance the accuracy. Under such perspective, this survey paper aims at providing a holistic literature review on how deep learning can make the transportation system more intelligent. More specifically, we categorize the applications in ITS that rely on an accurate learning model into visual recognition tasks, traffic flow prediction (TFP), traffic speed prediction (TSP), travel time prediction (TTP), and Miscellaneous tasks. We summarize the technology evolving of machine learning models, i.e., how traditional ML methods such as Support Vector Machine (SVM), Bayesian Network (BN) and Kalman Filter (KF) are used in the early stage, and afterwards, were revolutionized by the advent

* Corresponding author.
E-mail addresses: iseway@nus.edu.sg (Y. Wang), zhangdo@uestc.edu.cn (D. Zhang), daibo@uestc.edu.cn (B. Dai), leelh@nus.edu.sg (L.H. Lee).

https://doi.org/10.1016/j.trc.2018.12.004
Received 28 March 2018; Received in revised form 29 October 2018; Accepted 11 December 2018
Available online 21 December 2018
0968-090X/

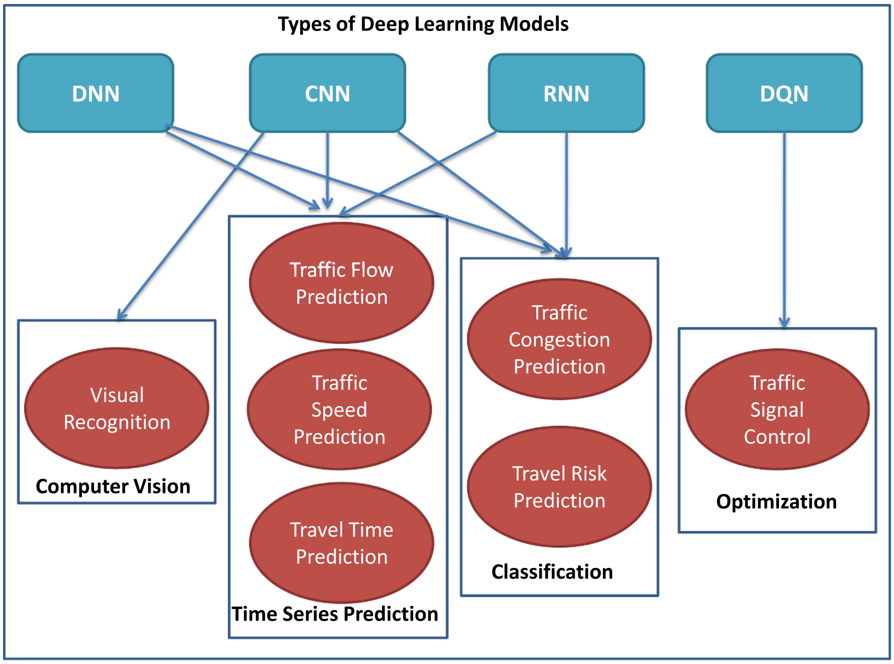


Fig. 1. Applications of deep learning models in ITS.

of various deep learning models.

To the best of our knowledge, this is the first paper that provides a comprehensive survey on the combination of intelligent transportation systems and deep learning. It puts a large number of related publications, that were originally scattered in diversified conferences and journals, in a self-consistent organization and provides the readers a clear picture of the technology evolving as well as the state-of-the-art deep models in the aforementioned applications of ITS. In our literature review, we also observe that there exist certain works solving the same problem with very similar models. Such redundant efforts can be avoided with the availability of this survey because it can help quickly justify the novelty and contribution of a piece of work that applies deep learning in an ITS application.

To sum up, readers can benefit from the survey in the following ways:

1. We provide a wide coverage on the applications of deep learning models in ITS. As shown in Fig. 1, DL models are mainly applied in four types of fundamental tasks, including computer vision, time series prediction, classification and optimization.
2. We carefully examine how various deep learning models are applied in each application and summarize the technology evolving trend. With such trend, it becomes easier for readers to identify the state-of-the-art models in a specific application. We also derive general and useful tips for the design of deep learning models.
3. We summarize the applicability, advantages and shortcomings of deep learning models in ITS. We also point out several future directions towards smarter transportation systems.

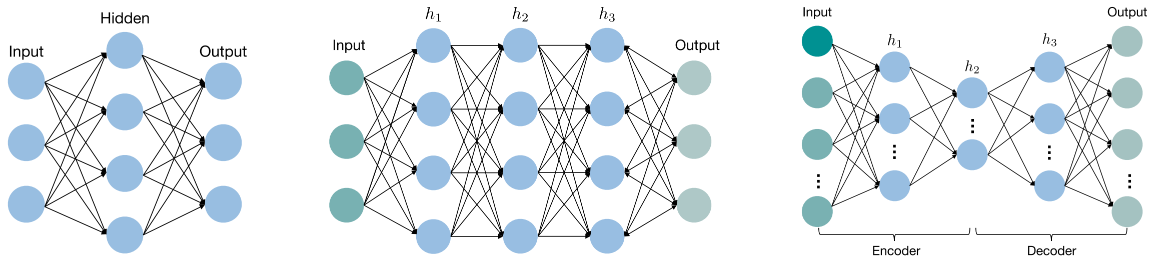
The remaining of the survey is organized as following. In Section 2, we provide the preliminary background of popular deep learning models that have been applied in ITS. In Section 3, we examine the problem of visual recognition tasks such as traffic sign recognition and summarize how deep learning models designed for computer vision are applied. In Sections 4.1–4.3, we review the deep models used to predict traffic flow, traffic speed and travel time, respectively. Readers will find that these three problems are defined in a similar way and can be solved by the same technique. In Section 5, we organize the remaining prediction problems in ITS as miscellaneous tasks. In Section 6, we discuss the applicability, advantages and shortcomings of deep learning models in ITS and provide the model design tips. Finally, we conclude the paper and present future directions in Section 7.

2. Preliminary background on deep learning

In this section, we present the preliminary background of deep learning models.

2.1. Deep neural network (DNN)

We follow (Min et al., 2016) to use deep neural network (DNN) to specifically refer to Multilayer Perceptron (MLP), Deep Belief Network (DBN) and Stacked Auto-Encoder (SAE), whose network structures are illustrated in Fig. 2. These models contain an input



(a) MLP Network (b) DBN Network (c) SAE Network

Fig. 2. Basic deep neural networks (DNN).

layer, one or multiple hidden layers and an output layer. At each layer, the input vector is multiplied by the weight matrix, whose parameters are to be learned, to produce the weighted sum. Then, a nonlinear function (also called activation function), such as *sigmoid*, hyperbolic tangent *tanh* or rectified linear function *ReLU* (Glorot and Bengio, 2011), is applied to generate the output of the hidden layer. The main differences between MLP, SAE and DBN lie in the design of the hidden layers.

An MLP is a feedforward artificial neural network with at least three layers (one input layer, one hidden layer and one output layer). Each node is fully connected to all the nodes in the following layer. The weight parameters in the hidden layer are trained in a supervised way with backpropagation (BP). It is recognized that MLP shows promising results when a sufficient amount of labeled data are available.

The network structure and training process of DBN and SAE are different from those in MLP. Their bottom layers are stacked with hidden variables for unsupervised parameter pre-training, with the goal of finding a good initial set of weights for the lower layers of the network. The parameters in these layers are trained in a layer-wise manner. In the top layer, there is a supervised classifier or predictor. Such design of network can reduce the amount of labeled data required for training.

In DBN, the stacked modules use Restricted Boltzmann machines (RBM) (Hinton and Salakhutdinov, 2006), a two-layered neural network with the lower layer called *visible* and the higher layer called *hidden*. The top two layers have undirected and symmetric connections between them, whereas the remaining lower layers receive top-down and directed connections from the layer above. In SAE, autoencoder (AE) (Hinton and Zemel, 1993) is used as the hidden layer. Its objective is to minimize the reconstruction error, i.e., it is trained to output the same values as the input vectors. The input is first encoded by the hidden layer and then decoded by the output layer. The number of units in the input and output layers is identical and the encoding and decoding operations are inverse to each other in a trained network.

In recent years, we have witnessed the application of DBN in speech recognition (Hinton et al., 2012), audio classification (Lee et al., 2009) and natural language understanding (Sarikaya et al., 2014); stacked auto-encoder in object recognition (Maria et al., 2016), image compression (Theis et al., 2017) and video retrieval (Song et al., 2018). RBM was also frequently used in face recognition (Teh and Hinton, 2000) and cross-media search (Wang et al., 2014, 2016). Note that there are a large number of relevant works that apply DL in various applications. Here, we only illustrate a few of them since our focus in this paper is deep learning applied in transportation-related applications.

2.2. Convolutional Neural Network (CNN)

Convolutional Neural Networks achieved great success for image classification in the competition of ImageNet (Krizhevsky et al., 2012) and has been later widely adopted in the applications of video classification (Karpathy et al., 2014), action recognition (Wang et al., 2018) and sentence classification (Kim, 2014). A typical CNN model starts from a convolutional layer, whose goal is to extract common patterns throughout the training instances. It consists of multiple kernel filters that are applied to the entirety of the image and transform the raw pixel values into higher level patterns. Each filter is in the form of a kernel matrix, whose entries are the parameters to be learned. The convolution operator involves taking patches from the input image of size equal to that of the kernel and calculating the dot product between the values in the path and those in the kernel matrix. The patch selection will be slided across the entire image, each step by a certain number of pixels, which is called stride. Consequently, the kernels are convolved with the input traffic image to obtain features ready for activation. Typically, each convolutional layer is followed by a non-linear activation function that allows the network to learn non-linear decision boundaries and indicate the image regions to be activated.

The output of the activation function in the convolutional layer is passed to the subsequent pooling layer with the objective of aggregating the information and reducing the representation. More specifically, the pooling layer takes a sliding window on its input matrix from the convolution layer and there is a $pool(\cdot)$ operator to either take the maximum value from the sliding window (called *max pooling*) or take the average of the values in the window (called *average pooling*). It is noticeable that max pooling is more frequently used as it provides a form of translation invariance and also captures the strong activation values. However, max pooling may lead to overfitting and there have been certain efforts to resolve the issue (Zeiler and Fergus, 2013; Kalchbrenner et al., 2014). We can see from Table 2 that almost all the CNN models for traffic sign recognition use max pooling.

The bundle of convolution layer and pooling layer can repeat multiple times to generate a deeper neural network. For example, Fig. 3 shows a CNN architecture with two convolution layers and two pooling layers. With more layers in the CNN network, higher-

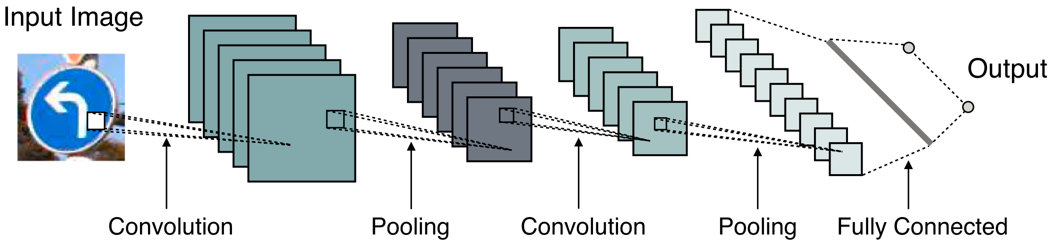


Fig. 3. Model architecture of convolutional neural network.

level information can be captured. The drawback is that there involve more parameters, which may lead to over-fitting. We can see in Table 2 that normally two or three convolution layers are used in the small-scale GTSRB dataset.

Finally, the small-size feature maps from the multiple kernel matrices and activation functions are concatenated into a long vector and passed to a fully connected layer. This layer typically used in the last stage of a CNN network and generates the desired number of outputs. The dimension of the output is equal to the number of classes in a classification task. Each dimension represents the probability of the input image belonging to the corresponding class.

2.3. Recurrent Neural Network (RNN)

Recurrent neural networks (RNNs) are specifically designed to model sequence data and have been applied in a wide range of applications such as speech recognition (Graves and Jaitly, 2014), machine translation (Sutskever et al., 2014), text generation (Sutskever et al., 2011) and video captioning (Gao et al., 2017). As shown in Fig. 4, its memory cells are chained and they perform the same task for every element in the input sequence. It is powerful because it can directly learn the mapping between the input and output sequences.

However, the traditional RNNs suffer from the problem of gradient vanishing or exploding. To resolve the issue, Long Short-Term Memory (LSTM) networks are proposed. Its memory cells or blocks chained are explicitly designed to maintain state over time and learn long-term dependencies. As shown in Fig. 5(a), each memory cell is enhanced with an explicit state vector and three types of non-linear gates, namely input gate, output gate and forget gate, to regulate the flow of signals into and out of the cell. Formally, let $X = (x_1, x_2, ..., x_N)$ denote the inputs of LSTM cell unit, where x_t is the input vector at the time-step t. Let σ be the sigmoid function and $\odot$ be element-wise multiplication. We have

$$\begin{aligned} i_t &= \sigma(\mathrm{W}^i x_t + \mathrm{V}^i h_{t-1} + \mathrm{b}^i) \\ f_t &= \sigma(\mathrm{W}^f x_t + \mathrm{V}^f h_{t-1} + \mathrm{b}^f) \\ o_t &= \sigma(\mathrm{W}^o x_t + \mathrm{V}^o h_{t-1} + \mathrm{b}^o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(\mathrm{W}^c x_t + \mathrm{V}^c h_{t-1} + \mathrm{b}^c) \\ h_t &= o_t \odot \tanh(c_t) \end{aligned}$$

where W^i, W^f, W^o, W^c are the weighted matrices for input gates, forget gates, output gates and cell state respectively, and b^i, b^f, b^o, b^c are their associated biases. These parameters are to be learned in the training stage. h_t is the vector of hidden layer which would be fed to the next cell unit as input.

The Gated Recurrent Unit (GRU) (Cho et al., 2014) is a simplified variant of the LSTM, without having separate memory cells. It still preserves the feature of resistance to the vanishing gradient problem, but is faster to train. The gates for a GRU cell are illustrated in Fig. 5(b). The LSTM cell has three gates, but the GRU cell only has two gates, one is update gate z and the other is reset gate r. The update gate determines the amount of memory to retain and the reset gate determines the amount of information from previous computed state to take into account. When r is close to 0, the reset gate would forget the previous state and act as if this is the first symbol of an input sequence.

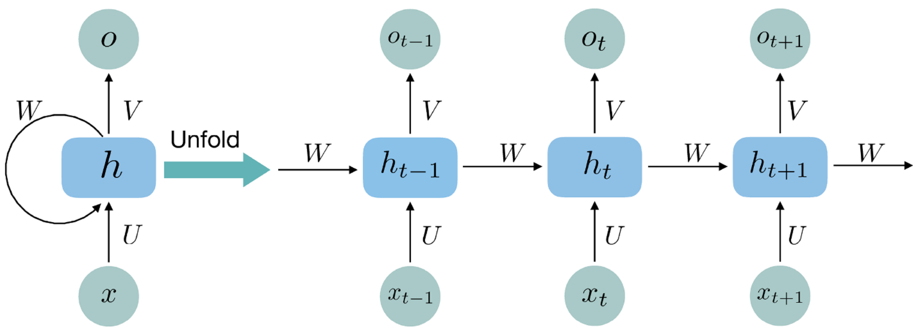
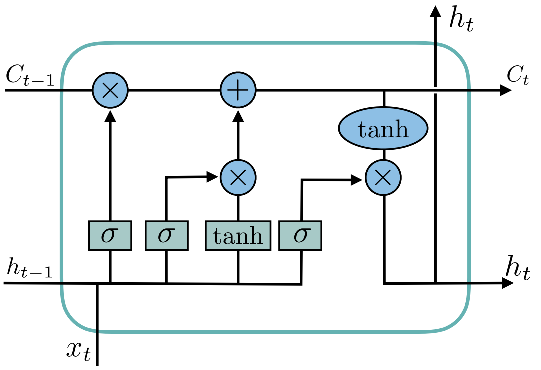
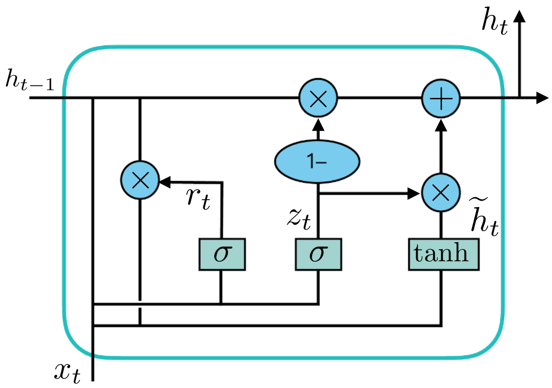


Fig. 4. Recurrent Neural Network (RNN).



(a) LSTM Network



(b) GRU Network

Fig. 5. Variants of RNNs.

$$z = \sigma(\mathrm{W}_z h_{t-1} + \mathrm{U}_z x_t) \tag{1}$$

$$r = \sigma(\mathrm{W}_r h_{t-1} + \mathrm{U}_r x_t) \tag{2}$$

$$c = \tanh(\mathrm{W}_c(h_{t-1} \odot r) + \mathrm{U}_c x_t) \tag{3}$$

$$h_t = z \odot c + (1 - z) \odot h_{t-1} \tag{4}$$

The aforementioned RNN, LSTM and GRU have their own counterparts of bi-directional variants, with similar underlying principle. The neurons of a regular network is split into two directions, one for forward direction and the other for backward direction. Since it involves more number of cells, the bidirectional variants are associated with more parameters to train and have higher potential in terms of expressiveness. The training process of the bi-directional networks is also similar to the unidirectional version. The back-propagation algorithm can still be applied because the two directional neurons do not have any interactions.

2.4. Deep reinforcement learning

In reinforcement learning (RL), given a set of internal states $S = s_1, ..., s_m$ and actions $A = a_1, ..., a_n$, the agent iteratively takes action a at state s and moves to a new state s' until a termination condition is satisfied. This process is guided by certain policies or rules π learned by interacting with the environment E. Rewards will be given to positive actions and we measure the true value of an action in state s as

$$Q_\pi(s, a) = \mathbb{E}[R_1 + \gamma R_2 + ...|S_0 = s, A_0 = a, \pi]$$

where $\gamma \in [0, 1]$ is a discounted factor for future rewards. The objective is to maximize the expected sum of future rewards through a sequence of actions.

Q-Learning is model-free and learns an optimal action-value function $Q(s, a)$. However, Q-value function needs to trace all possible state-action pairs, which is prohibitive. Mnih et al. introduced the Deep Q-Network (DQN) (Mnih et al., 2015) to approximate the Q-value function with a non-linear multi-layer convolutional network. Given state s, DQN outputs a vector of action values $Q(s, \cdot;\theta)$, where θ are the parameters of the network. For an m-dimensional state space S and an action space A containing n actions, the neural network serves as a function from $\mathbb{R}^m$ to $\mathbb{R}^n$. An experience replay memory is maintained to store previous transitions. In the mini-batch training stage, the observed transitions are sampled uniformly from the memory bank to update the network.

Due to its generality and robustness for sparse datasets, deep reinforcement learning has been successfully applied to solve a wide range of problems, including playing text-based games (Narasimhan et al., 2015), information extraction (Narasimhan et al., 2016), text generation (Guo, 2015), math word problem solving (Wang et al., 2018) and object detection in images (Caicedo and Lazebnik, 2015). In this paper, we will review how deep reinforcement learning is applied in ITS.

3. Visual recognition tasks in ITS

Traffic sign recognition, a very representative visual recognition task in ITS, plays a fundamental role in autonomous vehicles and advanced driver assistance systems (ADAS) (McCall and Trivedi, 2006; Schmidt et al., 2016). Its objective is to leverage computer vision techniques to automatically identify the correct signals from the traffic sign board. There is a relevant task named traffic sign detection (TSD), whose objective is to identify a region that refers to a traffic sign from a scene image. The accuracy of TSD is measured by mean Average Precision (mAP). To determine whether a detected region is a positive hit, we can calculate the value of IoU (intersection over union), i.e., $\frac{r \cap g}{r \cup g}$, and compare it with a threshold, which is normally set to 0.5 (Li et al., 2016; Qian et al., 2015). In contrast, TSR is an image classification problem and its accuracy refers to the portion of images that are correctly classified. The performances of these two tasks are often measured separately. Fig. 6 depicts a number of sample images from the benchmark

Fig. 6. Samples of traffic sign images in GTSRB.

dataset GTSRB that contains 51,839 images in 43 classes. The task is not trivial due to the challenging factors such as viewpoint variations, lighting conditions, motion blur, partial occlusion, color distortion, contrast degradation, etc. In the following, we first provide a brief review on how traditional machine learning methods solve the problem with hand-crafted features. Then, we put our emphasis on how deep learning models are adopted to improve the recognition accuracy.

Since traffic sign recognition is essentially a pattern recognition task, its accuracy mainly lies on the feature extractor as well as the classifier. Various types of discriminative and representative features have been adopted for traffic sign recognition. These features are expected to be robust for image rotation, translation and illumination variation. For example, Histogram of Oriented Gradient (HOG) and Haar-wavelets were used in (Ruta et al., 2010; Dalal and Triggs, 2005; Liu et al., 2014). SIFT (Lowe, 2004) is designed to keep local invariant characteristics and achieves a good result in Lowe (1999). As to the classifier, most methods tend to adopt the classic and general classification algorithms. For instances, SVM is the most prevalent model in TSR and has been used in Lafuente-Arroyo et al. (2005), Greenhalgh and Mirmehdi (2012), Maldonado-Bascón et al. (2007) and Le et al. (2010). Other classifiers like k-d trees and random forests are used for feature classification in Zaklouta et al. (2011) and Zaklouta and Stanciulescu (2012), respectively. In Ruta et al. (2010), boosting is proposed to improve the recognition performance by learning a set of weak classifiers and fusing a stronger one. Table 1 summarizes the experimental results on the benchmark dataset GRSBR by traditional machine learning techniques. The highest accuracy they can achieve is around 97%. In the following, we will show that by applying deep learning models, the accuracy can reach up to 99.84%, owning to the effective and automatic feature extraction.

Deep learning started to attract great attention in both academic and industry sectors after its great success in the ImageNet contest (Krizhevsky et al., 2012), a challenging image classification task with 1000 classes and 1.2 million high-resolution images. Since TSR is essentially an image classification problem, it is natural to transplant these models to the domain of TSR and avoid the cumbersome job to craft features manually. Hence, it is not surprising to find that the entire line of literature (Ciresan et al., 2011, 2012; Sermanet and LeCun, 2011; Jin et al., 2014; Haloi, 2015; Qian et al., 2015; Changzhen et al., 2016; Li and Yang, 2016; Li et al., 2016; Zeng et al., 2017; Jung et al., 2016; Zhang et al., 2017) uses convolutional neural network (CNN) or its variant as the deep learning model for the task of traffic sign recognition since CNN has witnessed great success in the task of image classification. In the following, we introduce the preliminary knowledge of CNN and provide an extensive comparison of its variants when applied in TSR.

In the year of 2011, Ciresan et al. (2011) and Sermanet and LeCun (2011) are the two first works to apply CNN in the area of traffic sign recognition. The only modification of the ConvNet structure in Ciresan et al. (2011) is to replace the max-pooling with the average-pooling. In Sermanet and LeCun (2011), a multi-scale CNN with two stages is proposed. Each stage is composed of a convolutional layer, a non-linear transform layer and a pooling layer. Its uniqueness lies in feeding the output of the first layer to the classifier as well. The motivation for combining the learned features of two stages is that the first stage extracts local motifs while the

Table 1
Summarization of accuracies on traditional (non-DL) machine learning methods.

Paper	Method	Dataset	Accuracy
Rajesh et al. (2011)	Simple neural network	GTSRB	94.73%
Boi and Gagliardini (2011)	SVM	GTSRB	96.89%
Zaklouta and Stanciulescu (2012)	k-d trees and random forests	GTSRB	97.2%

second stage captures the global and invariant shapes and structures.

In Ciresan et al. (2012), a pre-processing stage, including image translation, rotation and scaling, is incorporated to avoid overfitting and improve the generalization performance. Moreover, multiple CNN models are trained and their outputs are averaged as the learned feature. To improve the training time, Jin et al. Jin et al. (2014) propose to use hinge loss stochastic gradient descent (HLSGD) to train the CNN network. With hinge loss, a training example that is not helpful can be identified and there is no cost spent on the backpropagation (BP) to update the model parameters. It also brings the side product of improving the model generalization and its accuracy (99.65%) outperforms the previous competitors on the GTSRB dataset. To make the recognition robust to image translation, rotation and scaling, a spatial transformer layer is proposed in Haloi (2015). The layer is composed of three components: a localisation network, a grid generator and a sampling unit and supports operations such as translation, rotation, skew and cropping in the input feature map. Moreover, a modified GoogLeNet (Szegedy et al., 2015) is introduced in Haloi (2015) as the inception module. It uses varied size of convolutional filters to better capture features of different abstraction.

Qian et al. (2015) tackle the problems of traffic sign detection and recognition at the same time. In other words, a solution needs to first determine whether an image refers to a traffic sign. If yes, it further identifies the category. Technically, they use a variant of faster R-CNN (Ren et al., 2015) to produce candidate regions by RGB Space Thresholding. Subsequently, a traditional deep CNN model is trained to determine the category of a valid region proposal. By using a mixture of GTSRB and two other image datasets, MNIST (LeCun et al., 1998) and CASIA GB1 (Liu et al., 2011), that contain irrelevant images, the accuracies for traffic sign detection and recognition reach 97.56% and 98.83%, respectively. The work of Changzhen et al. (2016) also deals with the tasks of traffic sign detection and recognition simultaneously, but in another Chinese dataset harvested by the authors. The techniques are similar to those proposed in Qian et al. (2015), i.e., faster R-CNN for region proposal and deep CNN for sign recognition.

In Li and Yang (2016), Li et al. combine deep Restricted Boltzmann Machines (RBM) and Canonical Correlation Analysis (CAA) (Salakhutdinov and Hinton, 2009) for feature extraction. After the image preprocessing such as drizzling, gray-scale normalization and size normalization, local binary pattern (LBP) (Nosaka et al., 2011) are extracted. Then, these low-level features are transformed to high-level features through a two-layer RBM. After that, CAA is applied to determine the mapping relation among canonical variables. Finally, SVM is used to classify the derived feature vectors. Unfortunately, the accuracy of RBM-CAA on GTSRB dataset is only 96.68% and the performance is not as good as those using deep CNN models. Li et al. (2016) focuses on speed sign detection and recognition in a U.S. traffic sign dataset. It uses a modified R-CNN framework (Ren et al., 2015) to detect traffic signs, followed by a deep model with Cuda-convnet, that was proposed for Cifar-10 classification (Krizhevsky and Hinton, 2009; Torralba et al., 2008), to classify the traffic signs. Zeng et al. (2017) differs from previous methods that it employs a Lab-based perceptual color space (Perrot et al., 2014) rather than the traditional RGB color space. Experimental results show that the usage of such perceptual color space is more beneficial for CNN-based TSR.

The work of Jung et al. (2016) considers the real-time scenario of traffic sign detection that is useful for autonomous car driving. Its contribution is to avoid the expensive stage of selective search for region proposal. Instead, a simple color segmentation method is proposed to quickly generate the candidate regions. Experimental results show that the average fps (frames per second) can reach 16.9 Hz. Zhang et al. (2017) also considers fast traffic sign recognition in the CPU environment without using the expensive GPU hardware. A shallow network with three convolutional layers is used for feature extraction. Its activation function is replaced by ReLU Glorot and Bengio (2011) to improve computational efficiency. Different pooling operations are also leveraged to improve sign recognition performance. This model achieves so far the highest accuracy (99.84%) in the GTSRB dataset.

In Table 2, we summarize the batch of related works that apply CNN to solve the problem of TSR. For each method, we indicate whether it is designed for the task of sign detection. We report the number of ConvNet layers and the pooling strategy used in CNN, as well as the recognition accuracy on the dataset of GTSRB such that readers can easily identify the models that are superior. We can see that most of the CNNs proposed for TSR use two or three convolutional layers in the network design. This is mainly due to the scale of the dataset which contains only 51,839 images. The parameters in the shallow CNNs have been expressive enough to capture

Table 2
Summarization of deep learning models applied for traffic sign recognition.

Paper	Detection	Recognition	Conv layers	Pooling	Dataset	Accuracy
Ciresan et al. (2011)	×	√	2	max	GTSRB	99.15%
Sermanet and LeCun (2011)	×	√	2	max	GTSRB	99.17%
Ciresan et al. (2012)	×	√	2	max	GTSRB	99.46%
Jin et al. (2014)	×	√	3	max	GTSRB	99.65%
Haloi (2015)	×	√	2	max	GTSRB	99.81%
Qian et al. (2015)	√	√	3	max	GTSRB + MNIST + CASIA	99.83%
Changzhen et al. (2016)	√	√	–	max	Chinese traffic sign dataset	99%
Li and Yang (2016)	×	√	–	–	GTSRB	96.68%
Li et al. (2016)	√	√	3	max	LISA-TS (U.S. traffic signs)	97%
Zeng et al. (2017)	×	√	3	max	GTSRB	99.54%
Jung et al. (2016)	√	√	2	max	Korean-version traffic signs	–
Zhang et al. (2017)	×	√	3	max, average	GTSRB	99.84%

useful features. If the network is designed too deep, it may cause the problem of overfitting. We also observed that max-pooling is widely adopted among the CNNs. This is a common practice in the application of image recognition Krizhevsky et al. (2012). Compared with average-pooling, it is able to capture more distinctive features. Other tunable parameters in the neuron design include kernel size and stride. Since the images in TSR are relatively small, it is common to find that the kernel size is set to 3×3 and the stride is set to 1 or 2. From the table, we can see that the accuracy on GTSRB has reached 99.84%, with little room and meaning for further improvement unless a more challenging benchmark dataset is published.

Up to here, we have thoroughly reviewed the application of deep learning models for traffic sign recognition. There exist other computer-vision (CV) tasks that also apply deep learning models to improve the accuracy. Vehicle detection from images or surveillance videos is another representative example and a plenty number of related works have been published under the topic. According to the data sources, these works can be classified into vehicle detection for autonomous vehicles (Du et al., 2017; Lange et al., 2016; Wu and Lin, 2018), aerial images (Tang et al., 2017,; Zhong et al., 2017; Deng et al., 2017; Tayara et al., 2018; Qu et al., 2017), 3D range scan data (Li et al., 2016), low-resolution videos (Bautista et al., 2016), night-time video (Cai et al., 2016) and satellite images (Chen et al., 2014). They often adopt a two-step framework to solve the problem. First, a region proposal network such as Faster R-CNN (Ren et al., 2015) is adopted to identify possible regions that may contain vehicles. Note that there could be multiple vehicles within one image. In the second step, a deep neural network is trained for the verification of each candidate region. We observed that similar to the phenomenon in traffic sign recognition, convolutional network networks or the variants are dominating models that have been widely adopted. In addition, customizations were also developed for specific applications. For instance, real-time detection is a great concern in autonomous vehicles. To improve the efficiency, Lange et al. (2016) takes advantage of depth information from LIDAR sensors for region proposal and Wu and Lin (2018) detects vehicles based on motion clues, i.e., they set a fixed number of tracking points in a fixed location to detect moving vehicles. In contrast, the major concern of vehicle detection in aerial images is that the vehicles are too small to identify. A common strategy is to design hierarchical network to capture multi-scale image features. Examples include a deep convolutional neural network based on multi-scale spatial pyramid pooling proposed in Qu et al. (2017) and a vehicle proposal network based on hyper feature map proposed in Deng et al. (2017).

Another popular CV relevant task is the pedestrian detection which is useful for surveillance and automatic driving. Convolutional neural networks are still standard and effective solutions and widely adopted in the early works (Ouyang and Wang, 2013; Du et al., 2017; Fukui et al., 2015; John et al., 2015). Occlusion is one of the main challenges that many proposed models attempt to overcome. In Tian et al. (2015), each image is split into square cells and each cell is classified, with convolutional neural networks, into one part of body, such as shoulder, arms or legs. In Zhang et al. (2018), a simple and compact method based on the Faster R-CNN architecture was proposed to handle occlusion. Its contribution is building an attention mechanism across channels to represent various occlusion patterns. Some approaches utilize additional data sources to improve the detection accuracy. Schlosser et al. (2016) incorporates sensor data from LIDAR to construct a dense depth map and then extracts three features representing different aspects of the 3D scene. Liu et al. (2016) discovered that training thermal images with convolutional neural networks provides complementary information in discriminating human instances. To handle background clutter and large variations of pedestrian appearance, a switchable restricted Bolzmann machine was proposed in Luo et al. (2014) to explicitly model the visual variations at multiple levels. Recently, another network structure called scale-aware fast R-CNN has been proposed in Li et al. (2018) to handle the multi-scale issue. It applies multiple sub-networks to detect pedestrians from disjoint ranges and then adaptively combines them to generate the final detection results.

In the domain of automatic maintenance of transportation and civil infrastructure, deep learning models can be used to significantly reduce human intervention and operation cost. We observed that convolutional neural networks have been frequently used in tunnel inspection (Xue and Li, 2018; Makantasis et al., 2015), pavement crack detection (Zhang et al., 2017), cracks detection in concrete and steer surfaces (Cha et al., 2017), defect inspection of the catenary support devices (Chen et al., 2017), road damage detection (Zhang et al., 2016, 2018).

Besides vehicle detection and pedestrian detection, there are other sub-domains in autonomous driving that heavily rely on deep learning models. For instances, Ramos et al. (2017) used a fully convolutional network to detect small road hazards. End-to-end frameworks based on deep neural networks were proposed in Chen et al. (2017), Kim and Park (2017), Chen and Huang (2017) for lane detection and keeping the vehicle in the lane. In Maqueda et al. (2018), the problem of vehicle steering angle prediction was studied and solved by ResNet (Ren et al., 2015). In Bojarski et al. (2016), a bold idea was proposed to fully exploit the black-box functionality of deep learning models. More specifically, it trains a convolutional neural network to map raw pixels directly to steering commands. In Kim and Canny (2017), the attention model was applied to highlight the image regions to visually display the decision clues.

4. Traffic state prediction

There exists a number of traffic state prediction problems that have attracted intensive studies in the past decade. These tasks estimate different aspects of traffic state, such as traffic flow, travel speed and travel time. In this section, we review these prediction tasks that leverage deep learning models.

4.1. Traffic Flow Prediction (TFP)

Traffic flow prediction aims at estimating the number of vehicles flowing in/out a specific region or road segment within a future time window. It is obvious that the successful resolving of the task can benefit the dynamic traffic control, route planning, navigation services and other high-level applications.

In the formal definition, let us assume that each day is split into time intervals with fixed size such as 5 min. At the current time interval T, the task is to predict the traffic flow f_{T+1} based on the observed traffic flow information from the past time intervals $F = \{f_t | t = 1, 2, ..., T\}$. Some works may focus on predicting the traffic flow of the next several time intervals from $T + 1$ to $T + n$ as well. In practice, according to the length of projection time, traffic flow forecasts are divided into short-term (5–30 min), medium-term (30–60 min) and long-term (over an hour) (Yu et al., 2017). In the following, we briefly review the traditional approaches and then present how various deep learning models are applied to improve the accuracy. It is worth noting that there is no recognized benchmark dataset in this research area and authors tend to set up their own experimental environment for performance evaluation. The accuracy is normally measured by the errors between the predicted values and the ground-truth values. There are three types of measures commonly used:

$$MAE = \frac{1}{m}\sum_{i=1}^{m} |f_i - \hat{f}_i|$$

$$MRE = \frac{1}{m}\sum_{i=1}^{m} \frac{|f_i - \hat{f}_i|}{f_i}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m} |f_i - \hat{f}_i|^2}$$

Traffic flow prediction can be cast into the problem of time series analysis. The conventional methods can be divided into two categories: parametric methods and non-parametric methods. Autoregressive integrated moving average (ARIMA) model and its variants are the most prevalent parametric methods. Hamed et al. propose a simple ARIMA model of the order $(0, 1, 1)$ to predict the traffic flow on urban city (Hamed et al., 1995). A space–time ARIMA is proposed to forecast the traffic volume on urban areas in five-minute intervals in Ding et al. (2011). Kalman filter (KF) (Guo et al., 2014) is also widely adopted to address the TFP problem with smaller prediction errors. Hosseini et al. apply an adaptive fuzzy inference system based on KF to solve the nonlinear problem of traffic speed and flow forecasting (Hosseini et al., 2012). The aforementioned parametric approaches can achieve good performance when the traffic variations are regular. However, their forecast errors become remarkable under the irregular variations. To address the issue, non-parametric methods are proposed, such as k-nearest neighbor (k-NN) (Chang et al., 2012), non-parametric regression (Wu et al., 2012), support vector regression (SVR) (Castro-Neto et al., 2009). Non-parametric methods explore the latent routine from previous flow series data and automatically learn the weight. Chang et al. present a dynamic multi-interval traffic volume prediction model based on the k-NN nonparametric regression (Chang et al., 2012). In Wu et al. (2012), Wu et al. propose an online boosting regression technique that ensures traffic prediction under abnormal traffic conditions.

In the line of solvers by deep learning, there have been various types of models applied for traffic flow prediction. The most straightforward approach is to use three-layer feed-forward neural network, also named multi-layer perceptron (MLP). It is essentially a fully-connected network with a layer of hidden variables and the parameters can be learned via back-propagation (BP). Guo and Zhu (2009), Huang et al. (2013), Belay Habtie et al. (2015) and Akiyama and Inokuchi (2014) belong to this category. In Guo and Zhu (2009), Guo et al. simply apply MLP to predict the outgoing traffic of the multiple directions for a particular crossroad. This model is also used in Belay Habtie et al. (2015) to estimate traffic flow from data generated by cellular network. In Huang et al. (2013), a two-stage prediction framework is proposed. The historical traffic data are first divided into seven classes with fuzzy c-means clustering Cannon et al. (1986) and the i-th class is expected to capture the patterns for the i-th day of a week. Then, separate MLP models are trained with the data in each class. In Akiyama and Inokuchi (2014), MLP model is used to predict long-term traffic flow of urban expressways, with the purpose of measuring its correlation with national GDP growth.

A subsequent branch of related work (Huang et al., 2014; Koesdwiady et al., 2016; Lv et al., 2015; Leelavathi and Devi KJ, 2016; Yang et al., 2017) applies pre-training models such as deep belief network (DBN) and Stacked Auto-Encoder (SAE). More specifically, Huang et al. design a network with DBN at the bottom for unsupervised pre-training and a multi-task learning (MTL) layer for supervised prediction. Like traditional DBN, the parameters are trained in a greedy and layerwise manner with stacked restricted Boltzmann machines (RBMs). The top MTL layer is able to take full advantage of weight sharing in the DBN to provide better prediction results. Koesdwiady et al. (2016) further incorporate the weather conditions in the feature space. It also uses DBN for unsupervised pre-training but the top decision level is based on a fusion scheme to enhance prediction accuracy. As another type of pre-training model, SAE has also been attempted in Lv et al. (2015), Leelavathi and Devi KJ (2016), Yang et al. (2017). Among them, Lv et al. (2015) can be considered as the first work to apply auto-encoders as building blocks to represent traffic flow features. Similar to DBN, the parameters are learned in a greedy layerwise fashion with the purpose of capturing the spatial and temporal correlation inherently. On the top layer, a standard logistic regression model is used to accomplish the prediction task. In Leelavathi and Devi KJ (2016), we observe the same SAE model with logistic regression applied for traffic flow prediction. In Yang et al. (2017), Yang et al. propose to use Levenberg–Marquardt (LM) (Wilamowski and Yu, 2010) algorithm on top of the SAE network to replace logistic regression. The reason is that the authors believe LM can provide a numerical solution to the nonlinear problem minimizing a function over a space of the function parameters. In addition, it is stable and can generate good convergence Wilamowski and Yu (2010).

Finally, we discuss the approaches that rely on the more prevalent deep learning models including CNN, LSTM or their combination. Since LSTM has exhibited outstanding performance when handling sequence data, Fu et al. (2016) initiate a very simple attempt by directly applying LSTM for traffic flow prediction from the data collected by a specific sensor. In Yu et al. (2017), Yu et al. extend the network from one layer of LSTM to multiple layers of LSTM network for long-term traffic prediction. This deep LSTM

Table 3
Comparison of deep models for traffic flow prediction.

Paper	Year	Deep model	Future window	MAE on PeMS
Guo and Zhu (2009)	2009	MLP	Short term	
Huang et al. (2013)	2013	MLP	Short term	
Akiyama and Inokuchi (2014)	2014	MLP	Long term	
Belay Habtie et al. (2015)	2015	MLP	Short term	
Huang et al. (2014)	2014	DBN	Short term	
Koesdwiady et al. (2016)	2016	DBN	Short term	
Lv et al. (2015)	2015	SAE	Short term	
Leelavathi and Devi KJ (2016)	2016	SAE	Short term	34.1
Yang et al. (2017)	2017	SAE	Short term	
Fu et al. (2016)	2016	LSTM	Short term	17.21
Wu and Tan (2016)	2016	CNN + LSTM	Short term	19.37
Zhao et al. (2017)	2017	LSTM	Short term	
Wu et al. (2018)	2018	CNN + LSTM	Short term	19.13
Yu et al. (2017)	2017	LSTM	Long term	

model is further mixed with post-accident traffic patterns. In Zhao et al. (2017), Zhao et al. extend the LSTM network from the spatial dimension. Instead of training a separate LSTM model for each sensor as in previous work (Fu et al., 2016), they train a unique 2D LSTM network for all the sensors at the same time. The underlying intuition is to better capture the spatial relationship among these sensors in different road segments. Wu and Tan (2016) and Yu et al. (2017) are two approaches that combine the merits of CNN and LSTM networks. In Wu and Tan (2016), there is one CNN network to capture spatial features because the traffic flow of a location is dependent on its neighbors. Two additional LSTM networks, one for short-term features and the other for periodic features, are incorporated to mine short-term variability as well as periodicities of traffic flow. In the top layer, these features are concatenated and fed to a fully-connected layer for prediction. Recently, a hybrid model with CNN to mine spatial features and RNN to mine temporal features of traffic flow has also been proposed in Wu et al. (2018).

The methods that adopt deep learning models for the problem of TFP are summarized in Table 3. The publication year, deep model and the length of future prediction window are reported as the interesting fields. As to the details of network design, MLP, DBN and SAE are simple network and its main parameters are the number of layers and the number of hidden units within each layer. We observed that three layers of MLP, DBN and SAE are commonly used. The number of hidden units can be tuned according to the size of the dataset size. For instances, Huang et al. (2014) uses DBN with 128 hidden units in each layer and Koesdwiady et al. (2016) uses DBN with 250, 200 and 100 hidden units in the three layers, respectively. The parameters in the CNN and LSTM are not thoroughly discussed in the related papers of TFP. We only observe that they choose to use CNNs with three convolutional layers. As to LSTM, the number of hidden units is reported, but in great variance. This maybe due to the experiments were running on different validation datasets.

Due to the lack of a common benchmark dataset in this domain, it is difficult to directly justify the superiority of the deep models from the experimental comparison. We observed that the numeric experiments were conducted with datasets collected from various cities. A relatively popular dataset is Caltrans Performance Measurement System (PeMS).[1] It continuously collects loop detector data for more than 8100 freeway locations in California. The collected data are aggregated into 5-min interval for each detector station. We report the MAE measure of the related work that used PeMS in the numeric study. We can see that LSTM or CNN + LSTM perform better than SAE and reduce the MAE by 40–50%. Overall, it is generally recognized that CNN, LSTM are usually better than DBN and SAE, as we can see from the technology evolving trend in Table 3. As to the comparison with non-DL methods, DBN was reported in Koesdwiady et al. (2016) to reduce the MAE by more than 50% when compared with ARIMA. In Yu et al. (2017), LSTM was compared with regression approaches and the MAPE was reduced from more than 1.6 (by the regression methods) to 0.97 (by the LSTM models).

4.2. Traffic Speed Prediction (TSP)

The problem definition of traffic speed prediction (TSP) is in fact very similar to TFP. Given the historical traffic speed data of a road segment in the past time intervals, our objective is to predict its future speed in the subsequent time intervals. The known variable changes from the number of incoming/outgoing vehicles to the value of speed. Hence, we will see that these two problems adopt almost the same prediction strategy. In the literature, the solvers also start from ARIMA as the basic statistic model for time series prediction Ahmed and Cook (1979) and Duan et al. (2016). Other machine learning techniques such as Kalman Filter, SVM and Multivariate non-parametric regression are also observed in Okutani and Stephanedes (1984), Hong (2011), Clark (2003), respectively.

[1] http://pems.dot.ca.gov/.

The deep learning models in TSP follow a similar evolving trend to that in TFP because these two problems only differ in the variables used for prediction. The early approaches to solving TSP also start from multi-layer perceptron (MLP) network (Huang and Ran, 2003; Ye et al., 2012). To construct more informative feature vector, the outputs of neighboring road segments in the previous time intervals are used to capture spatial coherence of traffic flow in the road network (Ye et al., 2012).

As in TFP, an alternative category of deep models are based on pre-training such as SAE and DBN to reduce the number of required training samples. In the literature, SAE is directly adopted in Lemieux and Ma (2015), with MLP on the top layer as the predictor. Jia et al. (2016) apply DBN model, also with MLP as the predictor, for speed prediction. Their experimental results show that DBN outperforms MLP and ARIMA for all time horizons.

The last category applies CNN, LSTM or hybrid models. There are two different ways for CNN to extracted effective features from road networks. The first one is to treat the visualization of the whole map as an image input. The map contains the road network structure and the road segments may be plotted with different colors to represent the traffic state. The second one represents the road network or graph using a matrix of $n \times n$, where n is the number of nodes. Each matrix entry stores the status of a particular edge or road segment. Since road network is sparse and there are many empty entries in the matrix, a simplified representation is to use a vector of size $|E|$, where $|E|$ is the number of edges in the network. Wang et al. (2016) and Ma et al. (2017) are two approaches using CNN, but in different ways. In Wang et al. (2016), a recurrent CNN network is designed to predict the speed in multiple future time intervals simultaneously, i.e., the output of time $t + 1$ is used to predict speed at time $t + 2$. Ma et al. (2017) learns traffic information as an image. The input to the CNN network is a matrix of size $N \times Q$, where N is the length of past time intervals and Q is the length of road sections. Each entry m_{ij} is the average traffic speed on section i at time interval j. Since LSTM network and its variant are able to overcome the issue of back-propagated error decay through memory blocks, they are also applied in Ma et al. (2015) and Cui et al. (2018). In particular, a bidirectional LSTM layer is exploited to capture both spatial features and temporal dependencies from historical data. Recently, there have been some hybrid models proposed to improve the prediction accuracy. Yu et al. (2017) integrates one CNN and two LSTM networks, but in a way different from Wu and Tan (2016). In Wu and Tan (2016), the networks are located on the same layer and their output features are concatenated, whereas in Yu et al. (2017) the networks are stacked in a hierarchical structure and the output features of CNN are used as input for the subsequent LSTM network. The prediction module only relies on the output of the LSTM network in the top layer. In Jia et al. (2017), environmental factors such as rainfall is taken into account in the deep learning models. Both DBN and LSTM are investigated and results show that LSTM achieve superior performance in speed prediction. Another recent hybrid model was proposed in Li et al. (2017), which is called diffusion convolutional recurrent neural network (DCRNN). It captures the spatial dependency with random walks on the road network, and the temporal dependency with LSTM. To reduce the training time of LSTM, a spatio-temporal graph convolutional neural network (ST-GCNN) is proposed in Yu et al. (2017) for long-term traffic foresting tasks. Experimental results show that there is a $10\times$ acceleration of training speed.

Table 4 summarizes the methods in TSP. The model categorization and evolving trend look close to those in Table 3 because these two problems are essentially very similar. The neural network design does not have much difference with the previous models for TSR and TFP. In Ma et al. (2017), CNN with three convolutional layers is used and the number of units is set to 256, 128 and 64, respectively. In Yu et al. (2017), three-layer CNN is also used, but with the number of hidden units set to 32, 64 and 128, respectively. As to the datasets, we observed that a number of related work claimed to use traffic data from Beijing, which may be constructed from sources like taxi companies (Ma et al., 2017) or traffic management authority (Jia et al., 2016). Unfortunately, these datasets are not public and it is difficult to judge whether they are identical. For readers' reference, we can only report that the RMSE on their experimental study is roughly in the range of [5, 6.5]. It is also worth noting that the preparation of training and test datasets in TFP and TSP is different from that in TSR. Since TSR is an image classification problem, k-fold validation is normally used to obtain the accuracy. In TFP and TSP, the datasets are normally split into two periods, one as the historical data for training and the other as the future data for prediction. The purpose is to simulate the real scenario in which we use the historical data that have been collected to predict the unknown future.

Table 4
Comparison of deep models for traffic speed prediction.

Paper	Year	Deep model	Future window
Ye et al. (2012)	2012	MLP	Short term
Huang and Ran (2003)	2003	MLP	Short term
Lemieux and Ma (2015)	2015	SAE	Short term
Jia et al. (2016)	2016	DBN	Short term
Wang et al. (2016)	2016	CNN	Short term
Ma et al. (2017)	2017	CNN	Short term
Ma et al. (2015)	2015	LSTM	Short term
Cui et al. (2018)	2017	LSTM	Short term
Jia et al. (2017)	2017	LSTM	Short term
Liu and Chen (2017)	2017	SAE + DNN	Short term
Yu et al. (2017)	2017	CNN + LSTM	Short term
Yu et al. (2017)	2017	ST-GCNN	Long term

Table 5
Comparison of deep learning models for travel time prediction.

Paper	Year	Deep model
Park and Rilett (1999)	1999	MLP
Chen et al. (2004)	2004	MLP
Innamaa (2005)	2005	MLP
Li and Chen (2013)	2013	MLP
Gang et al. (2015)	2015	SAE
Siripanpornchana et al. (2016)	2016	DBN
Zeng and Zhang (2013)	2013	RNN
Liu et al. (2006)	2016	RNN
Duan et al. (2016)	2016	RNN
Jindal et al. (2017)	2017	RNN

4.3. Travel Time Prediction (TTP)

Travel Time Prediction (TTP) estimates the expected travel time across a road segment based on historical data. The problem itself is very close to TFP and TSP that have been examined in the previous two sections.

A straightforward approach is to apply the aforementioned speed prediction models to obtain the speed in the future time interval. Then, we can derive the travel time with the information of road segment length. An alternative solution is to directly apply the prediction models if the historical travel time information is available. Due to the space limit and the similarity of the problem definition of TTP to that of TFP and TSP, we present a very brief review of the deep learning models used for TTP and summarize the models in Table 5. Without much surprise, MLP has been the very early deep models applied to solve traffic prediction problems. Studies that apply MLP include (Park and Rilett, 1999; Chen et al., 2004; Innamaa, 2005; Li and Chen, 2013), among which a three-layer feed-forward neural network is normally used. Subsequently, there appear several works (Gang et al., 2015; Siripanpornchana et al., 2016) that apply pre-training models like DBN or SAE to improve the prediction accuracy. Recurrent neural networks such as state-space model (Van Lint et al., 2002, 2005; Liu et al., 2006; Zeng and Zhang, 2013) and LSTM (Duan et al., 2016) are also popular choices in recent two years. To the best of our knowledge, we did not find CNN models applied for travel time prediction.

5. Miscellaneous tasks

There have been a plenty number of other interesting applications that have been studied in ITS. In this section, we examine the miscellaneous tasks such as traffic matrix prediction, congestion management, travel risk prediction and traffic signal control.

5.1. Traffic tensor prediction

Traffic tensor prediction (Qian et al., 2008; Toqué et al., 2016), which provides the volume of traffic that flows between each pair of source–destination in a road network, is one of them. Both Qian et al. (2008) and Toqué et al. (2016) adopt recurrent neural network (RNN) or its variant to solve the problem. A novel deep spatio-temporal residual network is proposed to study the crowd flow between functional regions in a city (Zhang et al., 2017; Zhang et al., 2018). de Brébisson et al. (2015) and Jindal et al. (2017) are two works that rely on pure taxi GPS points for prediction, one using MLP for taxi destination prediction (de Brébisson et al., 2015) and the other proposing ST-NN (Spatial–Temporal Neural Network) to predict the travel time of a query in the form of (origin, destination, time-of-day). These are useful services in taxi hailing companies like UBER and DIDI.

Another related application is travel demand forecasting which is important to the tourism industry and there are several related efforts (Law and Au, 1999; Law, 2000; Chen et al., 2012; Claveria and Torra, 2014; Dantas et al., 2000) adopt artificial neural network to solve the problem in popular tourism cities such as Tokyo and Hong Kong. In these works, the input data could be different as they may have different data sources. The neural network contains only one hidden layer and can be seen as a simple MLP model. In Yao et al. (2018), the problem of taxi demand prediction is studied. It is a real problem driven by Didi Chuxing, which is China's ride-hailing giant. The problem is well solved by multi-view spatial–temporal network, which is essentially a hybrid model of CNN and LSTM networks. A similar problem was studied in Ke et al. (2017) to forecast passenger demand under on-demand ride services and a spatio-temporal deep learning model was proposed. Such passenger demand prediction models is helpful for the optimization of bus and taxi scheduling (Wang et al., 2017).

5.2. Congestion management and travel risk prediction

Congestion management (Ma et al., 2015; Chen et al., 2016) and travel risk prediction (Chen et al., 2016; Ren et al., 2017; Sameen and Pradhan, 2017) are also interesting applications In Ma et al. (2015), Ma et al. attempt to combine the merits of RBM and RNN for large-scale transportation network congestion evolution prediction because both RBM and RNN models have the capability of predicting a temporal sequence. In Chen et al. (2016), the traffic condition is categorized into *unimpeded* condition, *slow* condition, and *impeded* condition. Then, a stacked LSTM network with multiple layers is designed to predict the patterns of traffic conditions. In Sun et al. (2017), Sun et al. studied the problem of non-recurring congestion (NRC) which is caused by accidents, road construction work or special events. They proposed DxNAT, a deep neural network for non-recurring congestion prediction and explanation. In particular, they map the traffic data into images and apply CNN as the classifier. In Fouladgar et al. (2017), both CNN and LSTM models are empirically examined for the task of short-term congestion prediction.

In Chen et al. (2016, 2017), Alkheder et al. (2017), the problem of traffic accident risk-level prediction is examined. Various of deep models are proposed, including MLP (Alkheder et al., 2017), SAE (Chen et al., 2016) and a stacked LSTM with 4 layers (Ren et al., 2017). In Sameen and Pradhan (2017), Sameen et al. predict the injury severity of traffic accidents once they occurred in a road segment and the problem is solved by applying LSTM network. SAE is also used in Hatri and Boumhidi (2018) to integrate the spatial and temporal correlations of traffic flow inherently so as to detect traffic incidents. Recently, Zhang et al. (2018) employs Deep Belief Network (DBN) and Long Short-Term Memory (LSTM) in predicting traffic accidents from social media data. They crawled 3 million tweet data and explored the accidents from the text information so as to identify the road segments with accident risk.

5.3. Abnormal event detection in surveillance videos

Abnormal event detection in transportation surveillance videos is an application in which deep learning achieves the state-of-the-art performance. In Xu et al. (2015), Fan et al. (2018), Sun et al. (2018), stacked auto-encoder is used to learn discriminative features of appearance, motion and their joint representations, which are classified by SVM to find abnormal events. Another branch of related works (Shao et al., 2016; Sabokrou et al., 2018; Sun et al., 2017) use CNN for feature extraction. Readers interested in this topic can refer to Tripathi et al. (2018) and Sindagi and Patel (2018) for surveys on this topic.

5.4. Traffic signal control

Last but not the least, an interesting direction emerges with the success of AlphaGo (Silver et al., 2017) which applies deep reinforcement learning (RL) for optimal decision making in a huge search space. The idea is to use deep neural networks such as MLP as the value function approximator when the agent's representation of the environment, or state space, becomes too large. In this way, there is no need to explicitly maintain the Q-table. In the domain of ITS, we have witnessed the successful application of deep RL for traffic signal control (Li et al., 2016; Van der Pol and Oliehoek, 2016; Gao et al., 2017; Genders and Razavi, 2016).

The problem of traffic signal control is to properly control the traffic lights so as to reduce vehicle staying time at the intersections in the long run. Given a reward function, the agent controls the status of traffic lights based on the current traffic flow conditions. The key components in the deep reinforcement learning framework include designing proper states, actions as well as a reward function, and choosing a deep learning model to approximate the Q-learning function. The parameters in the DQN are normally learned through a large number of simulations. Most the related work used the Simulation of Urban MObility (SUMO) (Krajzewicz et al., 2012) for traffic simulation. SUMO provides useful APIs and GUI view to model large road networks and has been open-sourced. In the following, we explain these related research efforts in details.

In Li et al. (2016), Li et al. proposed to use deep stacked autoencoders (SAE) neural network to estimate the optimal Q-values. The reward function is designed as the difference between the number of queued vehicles in the directions of west-east and north–south. However, their simulation environment is simple and may not work in practice, since turning left or right at the intersection is not allowed. Van der Pol and Oliehoek (2016) combines the Deep Q-learning algorithm with a coordination algorithm for a scalable approach to controlling coordinating traffic lights. Its state is defined as a binary matrix of the positions of vehicles on the lanes as well as the current light configuration. The Q-function is learnt using a variant of transfer planning (Oliehoek et al., 2013). In Gao et al. (2017), Gao et al. extract useful features from vehicle position, speed and traffic signal state and use them to represent the state in the Markov decision process. The action is to select and actuate traffic signals. Two-layer convolutional network is selected as the approximator of Q-learning function. They also adopt the experience replay and target network proposed in Mnih et al. (2015) to learn the optimal traffic signal control policy. Genders and Razavi (2016) adopts a framework similar to Gao et al. (2017). It also uses CNN to approximate optimal Q-values. The state contains the information of vehicle position matrix, vehicle speed matrix and the current traffic signal. Its reward is defined as the change in cumulative vehicle delay. Casas (2017) attempted to solve traffic signal control in the city-scale with thousands of vehicle detectors and traffic lights. They score travel speed and use it for state construction. The reward is set to be the difference between the speed score and the baseline, which refers to the score derived by a hypothetical simulation. To handle the huge state and action space, Deep Deterministic Policy Gradient (DDPG) is utilized to improve the stability of reinforcement learning with value function approximation.

6. Discussions and insights

In this section, we provide a summary on the applicability of deep learning models in ITS. In addition, we summarize the advantages and shortcomings of deep learning models. Finally, we present tips for the model design in particular applications.

6.1. Applicability

From the general perspective, machine learning models are mainly applied in two fundamental tasks: classification and regression. The former predicts one or multiple class labels for an input and the latter predicts a real value. In other words, the output of classification is in the form of discrete classes whereas the output of regression is a continuous quantity. As reviewed in the aforementioned sections, deep learning models have been mainly applied in the tasks of computer vision (e.g., traffic sign recognition), time series prediction (e.g., traffic flow prediction, traffic speed prediction and traffic time prediction), classification (e.g., traffic congestion prediction and travel risk prediction) and optimization (e.g., traffic signal control). Among the applications, traffic sign recognition essentially belongs to the task of classification as its objective is to identify the traffic sign of an image from a limited number of options. As long as the application can be formulated as a problem of classification, regression or MDP (Markov Decision Process), and a large amount of training data are available or can be harvested at low cost, we believe deep learning models can be applied.

6.2. Advantages of DL models

Deep learning has witnessed great success in the areas of computer vision, speech recognition, item recommendation and natural language processing. In transportation applications, as reviewed, they also achieve the state-of-the-art performance in multiple classification and prediction tasks. As long as there are sufficient amount of training data and the GPU resources are available, it is highly likely that deep learning models can outperform the traditional machine learning techniques. We have witnessed considerable accuracy improvement by DL models in the tasks of image classification and time series prediction. In traffic sign prediction, the state-of-the-art non-DL method with random forest achieves an accuracy of 97.2% Zaklouta and Stanciulescu (2012). With CNNs, the accuracy is boosted to 99.84% (Zhang et al., 2017). In the task of traffic flow prediction, DBN was reported in Koesdwiady et al. (2016) to reduce the MAE by more than 50% when compared with ARIMA. LSTM was also shown to be superior to regression based methods (Yu et al., 2017). In traffic speed prediction, the prediction error of LSTM is 5.95, much smaller than the errors of SVM, Kalman Filter and ARIMA, which are reported to be 15.3, 20.11, 19.98, respectively. Thus, we consider high accuracy as the main advantage of DL models.

6.3. Shortcomings of DL models

Deep learning has its limitations as well. It has specific requirements on the amount of data and computing resources. In certain applications with limited training samples, such as predicting the breakdown probability of MRT system, these models are not applicable. The mainstream deep learning frameworks such as Tensorflow and PyTorch have provided a convenient way to allow users to customize the running environment, i.e., whether running the model with CPU or GPU. Since GPU has hundreds or even thousands of parallel computing units, its processing speed is much faster than CPU. Unfortunately, it is also much more expensive than CPU. Thus, users may need to take into account the trade-off between computation time and financial budget. In some applications like face recognition for personal id verification in airports or railway stations, there is a strong demand for real-time processing. The service providers need to deploy a large cluster of machines equipped with GPU to handle the highly concurrent requests. For other applications where longer processing time is tolerable, the DL models can run with CPU environment to save operation cost.

Parameter tuning in DL is a painful but necessary step. Users need to determine a number of hyper-parameters such as the learning rate, mini-batch size, the number of layers, the number of hidden units in each layer. Since the training process is slow and may take a few days or even weeks when there are enormous training data, it requires great engineering experience and expertise in order to find proper hyper-parameters. As pointed by Sutskever et al. (2013), parameter initialization is also an issue that requires special attention. Moreover, the choice of activation function and regularization methods (L1 norm or L2 norm) could also make certain difference. Readers can refer to Teney et al. (2017) for practical tips and tricks when designing DL models for visual question answering.

Interpretability is another key weakness of DL networks. They can obtain high discrimination power in automatic feature extraction, but at the cost of low interpretability of their black-box representations. In recent years, there are increasing efforts working to use visualization tools to alleviate the problem. For example, attention mechanism (Zhang et al., 2018; Vaswani et al., 2017) is a popular practice in DL models to assign different weights to regions in an image or terms in a sentence. The regions or terms that are believed to be important by machines can be highlighted in the visualization and verified by human. Readers can refer to a recent survey (Zhang and Zhu, 2018) which addresses visual interpretability for deep learning.

Moreover, the techniques of deep reinforcement learning to be applied in the optimization problems are still at the early stage. There is no clear evidence that they can outperform existing optimization solutions with benchmarked experiments. On the other hand, smart scheduling and real-time optimization have been the cornerstone functions in ITS. It is unlikely that they will be replaced by deep learning in the near future.

6.4. General guideline for DL model application

In this survey, we have covered various transportation-related applications that use DL models. Generally speaking, CNN is the best choice for image classification which has been frequently used in traffic sign recognition, vehicle and passenger tracking, obstacle and lane detection, and video-based surveillance. LSTM and its variants such as GRU and Bi-directional LSTM are designed to process sequential data. As surveyed in preceding sections, they can achieve promising accuracies for time series prediction problems such as traffic flow prediction, traffic speed prediction and travel time prediction. In the current stage, DBN and SAE are not as popular as CNN and LSTM and they are often used in the unsupervised pre-training step for a good parameter initialization (Erhan et al., 2010; Wang et al., 2016). Deep reinforcement learning is recommended for optimization problems such as traffic signal control to minimize the total waiting time for the red lights (Gao et al., 2017). They can also be applied for vehicle routing problems (Nazari et al., 2018).

6.5. Tips for DL model design

We have reviewed the technology evolving trend for the tasks of time series prediction and observed that they follow similar patterns. Initially, simple DNN models are applied. Afterwards, CNN or LSTM models are used for improvement. Finally, hybrid models are proposed and achieve state-of-the-art performance. It is also well recognized that CNN models are particularly effective to process image data; LSTM and GRU are more effective in extracting useful features from sequential data. In certain scenarios, these two types of models can be integrated in an end-to-end network to improve the accuracy. Finally, attention mechanism has been shown very effective in many applications (Vaswani et al., 2017). It can be conveniently integrated with existing deep learning models. Unfortunately, we rarely observe the usage of attention mechanism in ITS and this is a direction that is worth exploration.

When there lacks sufficient amount of training data, overfitting is rather common in DL models because they are too many parameters that require training. To resolve the issue, a useful strategy is to apply dropout (Srivastava et al., 2014) that randomly ignores parameter update in certain neurons during the training phase. Another solution is to apply regularization with L1 or L2 norm on the weight parameters of the network.

7. Conclusions and future directions

In this survey, we provide a comprehensive literature review of how deep learning models are applied in various transportation applications. In particular, four types of applications have been intensively examined, including traffic sign recognition (TSR), traffic flow prediction (TFP), traffic speed prediction (TSP) and travel time prediction (TTP). The first application is an instance of image classification and the entire line of literature adopts CNN to solve it. The remaining three applications are essentially time series prediction with more complicated context information such as network structure and weather conditions. Various models have been attempted to provide accurate prediction. We also discuss a miscellaneous collection of interesting applications in ITS, including traffic signal control that relies on deep reinforcement learning. We also discuss the pros and cons of deep learning models and their applicability.

In the following, we also summarize the future directions of further combination between deep learning and intelligent transportation systems, from the perspectives of data, model, device and application:

- **Data**. In ITS, real-time data from a huge number of sensors has been continuously generated, posing great challenges to handle the large volume, velocity and variety of big transportation data. These data, including weather status, air condition such as PM2.5, two-dimensional GPS location, vehicle speed, bus and MRT boarding and alighting data are structured and can be stored in a relational database. There are also a variety of unstructured data, such as video surveillance data, related to transportation. Most of the data are controlled by the government and we can see that it becomes trending for the government to publish and share the data with research institutes to better exploit the value of such big data. Data cleaning and repairing is important because data scientists spend 60% of their time on cleaning and organizing data.[2] In the domain of traffic data imputation (Li et al., 2014), deep learning models have also been used in Duan et al. (2016) to fix the issue of missing data. In map-mapping with noisy GPS data, a fusion approach with multiple data sources was proposed in Hu et al. (2017). HoloClean (Rekatsinas et al., 2017) is a recent open-source system that uses machine learning for automatic data cleaning and repairing. Besides HoloClean, a recent vision paper (Thirumuruganathan et al., 2018) also addresses data curation with deep learning. We also expect more attention in the future to effectively leverage the heterogeneous data, i.e., combining the multi-source data, to create new applications and enhance functionality in ITS.
- **Model**. In this survey, we spend most of the sections to cover applications about image recognition and time series prediction. Such applications are common and useful in ITS. In addition, they are not complex and can be normally solved by directly applying the existing DL models. In the future, we expect more complicated applications or tasks to emerge which would require more complex DL model design. For example, the convolutional network proposed for image classification has evolved from AlexNet (Krizhevsky et al., 2017), the first DL model that won the champion of the ImageNet compeition, to ResNet (He et al., 2016) which is able to train a very deep network with hundreds of layers. In the domain of autonomous driving, a number of complicated computer vision tasks could be involved and it could be an application to drive the demand for more powerful DL models.
- **Device**. In the related works examined in this survey, a client–server model is assumed, i.e, there is a remote server with supreme computing power. Thus, another future direction is to leverage the distributed computing power from the mobile chips in the client side. For example, the self-driving cars can detect the obstacles and correctly read the traffic signals. They can also form a network to communicate with each other. This requires compressing deep learning models or designing networks that are shallow but effective. These models that can be embedded in chips may become prevalent in the era of IoT and self-driving.
- **Application**. With the increasing data, computing resources and enhancement of deep models, more and more interesting

[2] https://blog.modeanalytics.com/python-data-cleaning-libraries/.

applications would become a reality. For instance, self-driving cars can be seen as a new application that integrates the techniques of computer vision and robotics. Social media data and transportation data can also be merged to create new applications. For example, Guo et al. studied to discover the passengers' personal interest by connecting their bus and MRT data with social media like Foursquare and Twitter (Guo et al., 2018; Zhang et al., 2018). The underlying idea is to annotate the regions with geo-tags. A user is more relevant to the associated tags if he/she frequently visits the region. With the derived personal interest, they formulate a new optimization problem of bus advertising with the objective of maximizing the influence effect on the passengers by the optimal assignment between ads and buses (Zhang et al., 2017; Guo et al., 2017). The combination of deep learning techniques and driver behavior modeling is an emerging research direction worth attention. For example, personalized lane-changing and car-following models (Meng et al., 2011; Butakov and Ioannou, 2015; Zheng, 2014) have been well studied and solved by traditional machine learning models. Applying deep learning to further improve the accuracy would be an interesting research topic to explore.

In summary, with the driven of machine learning, big data analysis and more powerful computing resources, there is no doubt that the transportation systems would become smarter and smarter. We have seen that as the collection, storage and analysis of multi-source transportation data become easier and cheaper, new and interesting applications emerge. For example, Wang et al. (2017) was pioneering in the sense that it collected the real customer demand from millions of citizens in Singapore to optimize the bus scheduling in a data-driven manner. The objective is to reduce the bus operation cost without sacrificing user experience, i.e., the average waiting time at the bus stops. Such data-driven approaches can be extended to other large-scale optimization problems such as bus network design and bus stop allocation. Besides the public transportation, the philosophy also applies in the bike-sharing and car-sharing applications. After they have collected sufficient amounts of data, customer demand and driver supply can be predicted with high accuracy by deep learning (Yao et al., 2018). Then, with smart matching and dispatching models, the user experience can be significantly upgraded with the existing infrastructure resources. Autonomous driving is another key application that represents the future and heavily relies on deep learning. There have been great efforts devoted for Level 4 – which refers to completely self-driving. It requires deep learning, coupled with other sensoring technologies, to continuously capture and analyze the surrounding environment in real-time. In other words, the vehicle is required to understand the semantic meaning of the traffic-related information. From the bigger perspective, when the self-driving vehicles have become safe enough and can be massively produced, we will face the ever largest car network whose communication and scheduling mechanisms would be rather challenging and may need to be reshaped. Hence, there is great room for technology evolving and we expect to witness more and deeper combination between deep learning and transportation systems.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under grants No. 61602087, the Fundamental Research Funds for the Central Universities under grants No. ZYGX2016J080, and A*STAR Industry Alignment Fund-Pre-Positioning Programme A1895a0033.

References

Ahmed, Mohammed S., Cook, Allen R., 1979. Analysis of freeway traffic time-series data by using Box-Jenkins techniques, Number 722.
Akiyama, Takamasa, Inokuchi, Hiroaki, 2014. Long term estimation of traffic demand on urban expressway by neural networks. In: ISIS, pp. 185–189,.
Alkheder, Sharaf, Taamneh, Madhar, Taamneh, Salah, 2017. Severity prediction of traffic accident using an artificial neural network. J. Forecast. 36 (1), 100–108.
Bautista, Carlo Migel, Dy, Clifford Austin, Mañalac, Miguel Iñigo, Orbe, Raphael Angelo, Cordel, Macario, 2016. Convolutional neural network for vehicle detection in low resolution traffic videos. In: TENSYMP. IEEE, pp. 277–281.
Biem, Alain, Bouillet, Eric, Feng, Hanhua, Ranganathan, Anand, Riabov, Anton, Verscheure, Olivier, Koutsopoulos, Haris N., Moran, Carlos, 2010. IBM infosphere streams for scalable, real-time, intelligent transportation services. In: SIGMOD, pp. 1093–1104.
Boi, Fabio, Gagliardini, Lorenzo, 2011. A support vector machines network for traffic sign recognition. In: IJCNN, pp. 2210–2216.
Bojarski, Mariusz, Del Testa, Davide, Dworakowski, Daniel, Firner, Bernhard, Flepp, Beat, Goyal, Prasoon, Jackel, Lawrence D., Monfort, Mathew, Muller, Urs, Zhang, Jiakai, Zhang, Xin, Zhao, Jake, Zieba, Karol, 2016. End to end learning for self-driving cars. CoRR, abs/1604.07316.
Butakov, V.A., Ioannou, P., 2015. Personalized driver/vehicle lane change models for adas. IEEE Trans. Veh. Technol. 64 (10), 4422–4431.
Caicedo, Juan C., Lazebnik, Svetlana, 2015. Active object localization with deep reinforcement learning. In: ICCV, pp. 2488–2496.
Cai, Yingfeng, Sun, Xiaoqiang, Wang, Hai, Chen, Long, Jiang, Hao-bin, 2016. Night-time vehicle detection algorithm based on visual saliency and deep learning. J. Sensors 2016 8046529:1–8046529:7.
Cannon, Robert L., Dave, Jitendra V., Bezdek, James C., 1986. Efficient implementation of the fuzzy c-means clustering algorithms. TPAMI 8 (2), 248–255.
Casas, Noe, 2017. Deep deterministic policy gradient for urban traffic light control. CoRR, abs/1703.09035.
Castro-Neto, Manoel, Jeong, Young-Seon, Jeong, Myong-Kee, Han, Lee D., 2009. Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. Exp. Syst. Appl. 36 (3), 6164–6173.
Cha, Young-Jin, Choi, Wooram, Büyüköztü rk, Oral, 2017. Deep learning-based crack damage detection using convolutional neural networks. CACAIE 32 (5), 361–378.
Chang, H., Lee, Youngjoo, Yoon, B., Baek, Sanghoon, 2012. Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. T-ITS 6 (3), 292–305.
Changzhen, Xiong, Cong, Wang, Weixin, Ma, Yanmei, Shan., 2016. A traffic sign detection algorithm based on deep convolutional neural network. In: ICSIP. IEEE, pp. 676–679.
Chen, Zhilu, Huang, Xinming, 2017. End-to-end learning for lane keeping of self-driving cars. In: IV, pp. 1856–1860.
Chen, Mei, Liu, Xiaobo, Xia, Jingxin, Chien, Steven I., 2004. A dynamic bus-arrival time prediction model based on apc data. CACAIE 19 (5), 364–376.
Chen, Chun-Fu, Lai, Ming-Cheng, Yeh, Ching-Chiang, 2012. Forecasting tourism demand based on empirical mode decomposition and neural network. Knowl.-Based Syst. 26, 281–287.
Chen, Xueyun, Xiang, Shiming, Liu, Cheng-Lin, Pan, Chunhong, 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. GRSL 11 (10), 1797–1801.

Chen, Yuan-yuan, Lv, Yisheng, Li, Zhenjiang, Wang, Fei-Yue, 2016. Long short-term memory model for traffic congestion prediction with online open data. In: ITSC. IEEE, pp. 132–137.
Chen, Quanjun, Song, Xuan, Yamada, Harutoshi, Shibasaki, Ryosuke, 2016. Learning deep representation from big and heterogeneous data for traffic accident inference. In: AAAI, pp. 338–344.
Chen, J., Liu, Z., Wang, H., Liu, K., 2017. High-speed railway catenary components detection using the cascaded convolutional neural networks. In: IST, pp. 1–6.
Chen, Shi-tao, Zhang, Songyi, Shang, Jinghao, Chen, Badong, Zheng, Nanning, 2017. Brain inspired cognitive model with attention for self-driving cars, CoRR, abs/1702.05596.
Cho, Kyunghyun, van Merrienboer, Bart, Gülçehre, Çaglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, Bengio, Yoshua, 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation, pp. 1724–1734.
Ciresan, Dan C., Meier, Ueli, Masci, Jonathan, Schmidhuber, Jürgen, 2011. A committee of neural networks for traffic sign classification. In: IJCNN, pp. 1918–1921.
Ciresan, Dan C., Meier, Ueli, Masci, Jonathan, Schmidhuber, Jürgen, 2012. Multi-column deep neural network for traffic sign classification. Neural Networks 32, 333–338.
Clark, Stephen, 2003. Traffic prediction using multivariate nonparametric regression. IJTE 129 (2), 161–168.
Claveria, Oscar, Torra, Salvador, 2014. Forecasting tourism demand to catalonia: neural networks vs. time series models. Econ. Model. 36, 220–228.
Cui, Zhiyong, Ke, Ruimin, Wang, Yinhai, 2018. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, volume abs/1801.02143.
Dalal, Navneet, Triggs, Bill, 2005. Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893.
Dantas, André, Yamamoto, Koshi, Lamar, Marcus V., Yamashita, Yaeko, 2000. Neural network for travel demand forecast using GIS and remote sensing. In: IJCNN (4), pp. 435–440.
de Brébisson, Alexandre, Simon, Étienne, Auvolat, Alex, Vincent, Pascal, Bengio, Yoshua, 2015. Artificial neural networks applied to taxi destination prediction.
Deng, Zhipeng, Sun, Hao, Zhou, Shilin, Zhao, Juanping, Zou, Huanxin, 2017. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. J-STARS 10 (8), 3652–3664.
Ding, Qing Yan, Wang, Xi Fu, Zhang, Xiu Yuan, Sun, Zhan Quan, 2011. Forecasting traffic volume with space-time arima model. In: Advanced Materials Research, vol. 156. Trans Tech Publ, pp. 979–983.
Duan, Peibo, Mao, Guoqiang, Zhang, Changsheng, Wang, Shangbo, 2016. Starima-based traffic prediction with time-varying lags. In: ITSC. IEEE, pp. 1610–1615.
Duan, Yanjie, Lv, Yisheng, Wang, Fei-Yue, 2016. Travel time prediction with lstm neural network. In: ITSC. IEEE, pp. 1053–1058.
Duan, Y., Lv, Y., Liu, Y.L., Wang, F.Y., 2016. An efficient realization of deep learning for traffic data imputation. Transp. Res. Part C: Emerg. Technol. 72, 168–181.
Du, Xinxin, Ang, Marcelo H., Rus, Daniela, 2017. Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework. In: IROS, pp. 749–754.
Du, Xianzhi, El-Khamy, Mostafa, Lee, Jungwon, Davis, Larry S., 2017. Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection. In: WACV, pp. 953–961.
Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron C., Manzagol, Pierre-Antoine, Vincent, Pascal, Bengio, Samy, 2010. Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11, 625–660.
Fan, Yaxiang, Wen, Gongjian, Li, Deren, Qiu, ShaoHua, Levine, Martin D., 2018. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. CoRR, abs/1805.11223.
Fouladgar, Mohammadhani, Parchami, Mostafa, Elmasri, Ramez, Ghaderi, Amir, 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction, pp. 2251–2258.
Fu, Rui, Zhang, Zuo, Li, Li, 2016. Using lstm and gru neural network methods for traffic flow prediction. In: YAC, pp. 324–328.
Fukui, Hiroshi, Yamashita, Takayoshi, Yamauchi, Yuji, Fujiyoshi, Hironobu, Murase, Hiroshi, 2015. Pedestrian detection based on deep convolutional neural network with ensemble inference network. In: IV, pp. 223–228.
Gang, Xiong, Kang, Wenwen, Wang, Feiyue, Zhu, Fenghua, Lv, Yisheng, Dong, Xisong, Riekki, Jukka, Pirttikangas, Susanna, 2015. Continuous travel time prediction for transit signal priority based on a deep network. In: ITSC. IEEE, pp. 523–528.
Gao, Juntao, Shen, Yulong, Liu, Jia, Ito, Minoru, Shiratori, Norio, 2017. Adaptive traffic signal control: deep reinforcement learning algorithm with experience replay and target network, arXiv preprint arXiv:1705.02755.
Gao, Juntao, Shen, Yulong, Liu, Jia, Ito, Minoru, Shiratori, Norio, 2017. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. CoRR, abs/1705.02755.
Gao, Lianli, Guo, Zhao, Zhang, Hanwang, Xu, Xing, Shen, Heng Tao, 2017. Video captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimedia 19 (9), 2045–2055.
Genders, Wade, Razavi, Saiedeh, 2016. Using a deep reinforcement learning agent for traffic signal control. CoRR, abs/1611.01142.
Glorot, Xavier, Bordes, Antoine, Bengio, Yoshua, 2011. Deep sparse rectifier neural networks. In: AISTATS, pp. 315–323.
Graves, Alex, Jaitly, Navdeep, 2014. Towards end-to-end speech recognition with recurrent neural networks. In: ICML, pp. 1764–1772.
Greenhalgh, Jack, Mirmehdi, Majid, 2012. Real-time detection and recognition of road traffic signs. ITS 13 (4), 1498–1506.
Guo, Hongyu, 2015. Generating text with deep reinforcement learning. CoRR, abs/1510.09202.
Guo, Xiaojian, Zhu, Quan, 2009. A traffic flow forecasting model based on bp neural network. In: PEITS, pp. 311–314.
Guo, Jianhua, Huang, Wei, Williams, Billy M, 2014. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. TRC 43, 50–64.
Guo, Long, Zhang, Dongxiang, Cong, Gao, Wu, Wei, Tan, Kian-Lee, 2017. Influence maximization in trajectory databases. IEEE Trans. Knowl. Data Eng. 29 (3), 627–641.
Guo, Long, Zhang, Dongxiang, Wang, Yuan, Wu, Huayu, Cui, Bin, Tan, Kian-Lee, 2018. Co^2: inferring personal interests from raw footprints by connecting the offline world with the online world. ACM Trans. Inf. Syst. 36 (3), 31:1–31:29.
Habtie, Ayalew Belay, Abraham, Ajith, Midekso, Dida, 2015. Cellular network based real-time urban road traffic state estimation framework using neural network model estimation. In: SSCI, pp. 38–44.
Haloi, Mrinal, 2015. Traffic sign classification using deep inception based convolutional networks. CoRR, abs/1511.02992.
Hamed, Mohammad M., Al-Masaeid, Hashem R., Said, Zahi M. Bani, 1995. Short-term prediction of traffic volume in urban arterials. IJTE 121 (3), 249–254.
Hatri, Chaimae El, Boumhidi, Jaouad, 2018. Fuzzy deep learning based urban traffic incident detection. Cogn. Syst. Res. 50, 206–213.
He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778.
Hinton, Geoffrey E., Salakhutdinov, Ruslan R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.
Hinton, Geoffrey E., Zemel, Richard S., 1993. Autoencoders, minimum description length and helmholtz free energy. In: NIPS, pp. 3–10.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.
Hong, Wei-Chiang, 2011. Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm. Neurocomputing 74 (12), 2096–2107.
Hosseini, Seyed Hadi, Moshiri, Behzad, Rahimi-Kian, Ashkan, Araabi, Babak N., 2012. Short-term traffic flow forecasting by mutual information and artificial neural networks. In: ICIT. IEEE, pp. 1136–1141.
Huang, Shan-Huen, Ran, Bin, 2003. An application of neural network on traffic speed prediction under adverse weather condition (Ph.D. thesis). University of Wisconsin–Madison.
Huang, He, Tang, Qifeng, Liu, Zhen, 2013. Adaptive correction forecasting approach for urban traffic flow based on fuzzy c-mean clustering and advanced neural network. J. Appl. Math. 2013, 195824:1–195824:7.
Huang, Wenhao, Song, Guojie, Hong, Haikun, Xie, Kunqing, 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. ITS 15

(5), 2191–2201.
Hu, Gang, Shao, Jie, Liu, Fenglin, Wang, Yuan, Shen, Heng Tao, 2017. If-matching: towards accurate map-matching with information fusion. IEEE Trans. Knowl. Data Eng. 29 (1), 114–127.
Innamaa, Satu, 2005. Short-term prediction of travel time using neural networks on an interurban highway. Transportation 32 (6), 649–669.
Jia, Yuhan, Wu, Jianping, Du, Yiman, 2016. Traffic speed prediction using deep learning method. In: ITSC, pp. 1217–1222.
Jia, Yuhan, Wu, Jianping, Ben-Akiva, Moshe, Seshadri, Ravi, Du, Yiman, 2017. Rainfall-integrated traffic speed prediction using deep learning method. T-ITS 11 (9), 531–536.
Jindal, Ishan, Chen, Xuewen, Nokleby, Matthew, Ye, Jieping, et al., 2017. A unified neural network approach for estimating travel time and distance for a taxi trip. arXiv preprint arXiv:1710.04350.
Jin, Junqi, Fu, Kun, Zhang, Changshui, 2014. Traffic sign recognition with hinge loss trained convolutional neural networks. ITS 15 (5), 1991–2000.
John, Vijay, Mita, Seiichi, Liu, Zheng, Qi, Bin, 2015. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In: MVA, pp. 246–249.
Jung, Seokwoo, Lee, Unghui, Jung, Jiwon, Shim, David Hyunchul, 2016. Real-time traffic sign recognition system with deep convolutional neural network. In: URAI, pp. 31–34.
Kalchbrenner, Nal, Grefenstette, Edward, Blunsom, Phil, 2014. A convolutional neural network for modelling sentences. In: ACL, pp. 655–665.
Karpathy, Andrej, Toderici, George, Shetty, Sanketh, Leung, Thomas, Sukthankar, Rahul, Li, Fei-Fei, 2014. Large-scale video classification with convolutional neural networks. In: CVPR, pp. 1725–1732.
Ke, Jintao, Zheng, Hongyu, Yang, Hai, Chen, Xiqun (Michael), 2017. Short-term forecasting of passenger demand under on- demand ride services: a spatio-temporal deep learning approach. Transp. Res. Part C: Emerg. Technol. 85, 591–608.
Kim, Yoon, 2014. Convolutional neural networks for sentence classification. In: ACL, pp. 1746–1751.
Kim, Jinkyu, Canny, John F., 2017. Interpretable learning for self-driving cars by visualizing causal attention. In: ICCV, pp. 2961–2969.
Kim, Jiman, Park, Chanjong, 2017. End-to-end ego lane estimation based on sequential transfer learning for self-driving cars. In: CVPR Workshops, pp. 1194–1202.
Koesdwiady, Arief, Soua, Ridha, Karray, Fakhreddine, 2016. Improving traffic flow prediction with weather information in connected cars: a deep learning approach. TVT 65 (12), 9508–9517.
Krajzewicz, Daniel, Erdmann, Jakob, Behrisch, Michael, Bieker, Laura, 2012. Recent development and applications of SUMO - Simulation of Urban MObility. Int. J. Adv. Syst. Meas. 5 (3&4), 128–138.
Krizhevsky, Alex, Hinton, Geoffrey, 2009. Learning multiple layers of features from tiny images.
Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey E., 2012. Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114.
Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey E., 2017. Imagenet classification with deep convolutional neural networks. Commun. ACM 84–90.
Lafuente-Arroyo, Sergio, Gil-Jiménez, Pedro, Maldonado-Bascon, R., López-Ferreras, Francisco, Maldonado-Bascón, Saturnino, 2005. Traffic sign shape classification evaluation i: Svm using distance to borders. In: IV, pp. 557–562.
Lange, Stefan, Ulbrich, Fritz, Göhring, Daniel, 2016. Online vehicle detection using deep neural networks and lidar based preselected image patches. In: IV, pp. 954–959.
Law, R., 2000. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. Tourism Manage. 21 (4), 331–340.
Law, R., Au, N., 1999. A neural network model to forecast japanese demand for travel to Hong Kong. Tourism Manage. 20 (1), 89–97.
LeCun, Yann, Bottou, Léon, Bengio, Yoshua, 1998. and Patrick Haffner. Gradient-based Learning Applied to Document Recognition, vol. 86 IEEE.
Lee, Honglak, Pham, Peter T., Largman, Yan, Ng, Andrew Y., 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: NIPS, pp. 1096–1104.
Leelavathi, M.V., Sahana Devi, K.J., 2016. An architecture of deep learning method to predict traffic flow in big data.
Le, Tam T., Tran, Son T., Mita, Seichii, Nguyen, Thuc D., 2010. Real time traffic sign detection using color and shape-based features. In: ACIIDS, pp. 268–278.
Lemieux, Joe, Ma, Yuan, 2015. Vehicle speed prediction using deep learning. In: VPPC. IEEE, pp. 1–5.
Li, Chi-Sen, Chen, Mu-Chen, 2013. Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks. NCA 23 (6), 1611–1629.
Li, Chen, Yang, Cheng, 2016. The research on traffic sign recognition based on deep learning. In: ISCIT, pp. 156–161.
Li, Y., Li, Z., Li, L., 2014. Missing traffic data: comparison of imputation methods. IET Intel. Transp. Syst. 8 (1), 51–57.
Li, L., Lv, Y., Wang, F.Y., 2016. Traffic signal timing via deep reinforcement learning. JAS 3 (3), 247–254.
Li, Yuan, Møgelmose, Andreas, Trivedi, Mohan Manubhai, 2016. Pushing the speed limit: high-accuracy US traffic sign recognition with convolutional neural networks. T-IV 1 (2), 167–176.
Li, Bo, Zhang, Tianlei, Xia, Tian, 2016. Vehicle detection from 3d lidar using fully convolutional network. In: Robotics: Science and Systems XII, June 18 - June 22, 2016. University of Michigan, Ann Arbor, Michigan, USA.
Li, Yaguang, Yu, Rose, Shahabi, Cyrus, Liu, Yan, 2017. Graph convolutional recurrent neural network: data-driven traffic forecasting. CoRR, abs/1707.01926.
Li, Jianan, Liang, Xiaodan, Shen, Shengmei, Xu, Tingfa, Feng, Jiashi, Yan, Shuicheng, 2018. Scale-aware fast R-CNN for pedestrian detection. IEEE Trans. Multimedia 20 (4), 985–996.
Liu, Lijuan, Chen, Rung-Ching, 2017. A novel passenger flow prediction model using deep learning methods. Transp. Res. Part C: Emerg. Technol. 84, 74–91.
Liu, Hao, Zuylen, Henk van, Lint, Hans van, Salomons, Maria, 2006. Predicting urban arterial travel time with state-space neural networks and Kalman filters. TRR (1968), 99–108.
Liu, Cheng-Lin, Yin, Fei, Wang, Qiu-Feng, Wang, Da-Han, 2011. ICDAR 2011 Chinese handwriting recognition competition. In: ICDAR, pp. 1464–1469.
Liu, Huaping, Liu, Yulong, Sun, Fuchun, 2014. Traffic sign recognition using group sparse coding. Inf. Sci. 266, 75–89.
Liu, Jingjing, Zhang, Shaoting, Wang, Shu, Metaxas, Dimitris N., 2016. Multispectral deep neural networks for pedestrian detection. In: BMVC.
Lowe, David G., 1999. Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157.
Lowe, David G., 2004. Distinctive image features from scale-invariant keypoints. IJCV 60 (2), 91–110.
Luo, Ping, Tian, Yonglong, Wang, Xiaogang, Tang, Xiaoou, 2014. Switchable deep network for pedestrian detection. In: CVPR, pp. 899–906.
Lv, Yisheng, Duan, Yanjie, Kang, Wenwen, Li, Zhengxi, Wang, Fei-Yue, 2015. Traffic flow prediction with big data: a deep learning approach. ITS 16 (2), 865–873.
Maeda, Hiroya, Sekimoto, Yoshihide, Seto, Toshikazu, Kashiyama, Takehiro, Omata, Hiroshi, 2018. Road damage detection using deep neural networks with images captured through a smartphone. CoRR, abs/1801.09454.
Ma, Xiaolei, Tao, Zhimin, Wang, Yinhai, Yu, Haiyang, Wang, Yunpeng, 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. TRC 54, 187–197.
Ma, Xiaolei, Yu, Haiyang, Wang, Yunpeng, Wang, Yinhai, 2015. Large-scale transportation network congestion evolution prediction using deep learning theory. PloS One 10 (3), e0119044.
Ma, Xiaolei, Dai, Zhuang, He, Zhengbing, Ma, Jihui, Wang, Yong, Wang, Yunpeng, 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17 (4), 818.
Makantasis, Konstantinos, Protopapadakis, Eftychios, Doulamis, Anastasios, Doulamis, Nikolaos, Loupos, Konstantinos, 2015. Deep convolutional neural networks for efficient vision based tunnel inspection. In: ICCP, pp. 335–342.
Maldonado-Bascón, Saturnino, Lafuente-Arroyo, Sergio, Gil-Jiménez, Pedro, Gómez-Moreno, Hilario, López-Ferreras, Francisco, 2007. Road-sign detection and recognition based on support vector machines. ITS 8 (2), 264–278.
Maqueda, Ana I., Loquercio, Antonio, Gallego, Guillermo, García, Narciso N., Scaramuzza, Davide, 2018. Event-based vision meets deep learning on steering prediction for self-driving cars. CoRR, abs/1804.01310.
Maria, Joao, Amaro, Joao, Falcao, Gabriel, Alexandre, Luís A., 2016. Stacked autoencoders using low-power accelerated architectures for object recognition in autonomous systems. Neural Process. Lett. 43 (2), 445–458.

McCall, Joel C., Trivedi, Mohan M., 2006. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. ITS 7 (1), 20–37.
Meng, F., Zhang, W., Wang, J., 2011. Driver's car-following and lane-changing models: an experimental study. In: 2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management Vol. Part 3. pp. 2045–2048.
Min, Seonwoo, Lee, Byunghan, Yoon, Sungroh, 2016. Deep learning in bioinformatics. CoRR, abs/1603.06430.
Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin A., Fidjeland, Andreas, Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, Hassabis, Demis, 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.
Narasimhan, Karthik, Kulkarni, Tejas D., Barzilay, Regina, 2015. Language understanding for text-based games using deep reinforcement learning. In: EMNLP, pp. 1–11.
Narasimhan, Karthik, Yala, Adam, Barzilay, Regina, 2016. Improving information extraction by acquiring external evidence with reinforcement learning. In: EMNLP, pp. 2355–2365.
Nazari, MohammadReza, Oroojlooy, Afshin, Snyder, Lawrence V., Takác, Martin, 2018. Deep reinforcement learning for solving the vehicle routing problem. CoRR, abs/1802.04240.
Nosaka, Ryusuke, Ohkawa, Yasuhiro, Fukui, Kazuhiro, 2011. Feature extraction based on co-occurrence of adjacent local binary patterns. In: PSIVT, pp. 82–91.
Okutani, Iwao, Stephanedes, Yorgos J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. TRB 18 (1), 1–11.
Oliehoek, Frans A., Whiteson, Shimon, Spaan, Matthijs T.J., 2013. Approximate solutions for factored dec-pomdps with many agents. In: AAMAS, pp. 563–570.
Ouyang, Wanli, Wang, Xiaogang, 2013. Joint deep learning for pedestrian detection. In: ICCV, pp. 2056–2063,.
Park, Dongjoo, Rilett, Laurence R., 1999. Forecasting freeway link travel times with a multilayer feedforward neural network. CACAIE 14 (5), 357–367.
Perrot, Michaël, Habrard, Amaury, Muselet, Damien, Sebban, Marc, 2014. Modeling perceptual color differences by local metric learning. In: ECCV, pp. 96–111.
Qian, Feng, Hu, Guangmin, Xie, Jijun, 2008. A recurrent neural network approach to traffic matrix tracking using partial measurements. In: ICIEA. IEEE, pp. 1640–1643.
Qian, Rongqiang, Zhang, Bailing, Yue, Yong, Wang, Zhao, Coenen, Frans, 2015. Robust Chinese traffic sign detection and recognition with deep convolutional neural network. In: ICNC, pp. 791–796.
Qu, Tao, Zhang, Quanyuan, Sun, Shilei, 2017. Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. Multimedia Tools Appl. 76 (20), 21651–21663.
Rajesh, Reghunadhan, Rajeev, K., Suchithra, K., Prabhu, Lekhesh V., Gopakumar, V., Ragesh, N.K., 2011. Coherence vector of oriented gradients for traffic sign recognition using neural networks. In: IJCNN, pp. 907–910.
Ramos, Sebastian, Gehrig, Stefan K., Pinggera, Peter, Franke, Uwe, Rother, Carsten, 2017. Detecting unexpected obstacles for self-driving cars: fusing deep learning and geometric modeling. In: IV, pp. 1025–1032.
Rekatsinas, Theodoros, Chu, Xu, Ilyas, Ihab F., Ré, Christopher, 2017. Holoclean: holistic data repairs with probabilistic inference. PVLDB 10 (11), 1190–1201.
Ren, Shaoqing, He, Kaiming, Girshick, Ross B., Sun, Jian, 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99.
Ren, Honglei, Song, You, Liu, JingXin, Hu, Yucheng, Lei, Jinzhi, 2017. A deep learning approach to the prediction of short-term traffic accident risk. arXiv preprint arXiv:1710.09543.
Ruta, Andrzej, Li, Yongmin, Liu, Xiaohui, 2010. Robust class similarity measure for traffic sign recognition. ITS 11 (4), 846–855.
Sabokrou, Mohammad, Fayyaz, Mohsen, Fathy, Mahmood, Moayed, Zahra., Klette, Reinhard, 2018. Deep-anomaly: fully convolutional neural network for fast anomaly detection in crowded scenes. Comput. Vision Image Underst.
Salakhutdinov, Ruslan, Hinton, Geoffrey E., 2009. Deep Boltzmann machines. In: AISTATS, pp. 448–455.
Sameen, Maher Ibrahim, Pradhan, Biswajeet, 2017. Severity prediction of traffic accidents with recurrent neural networks. Appl. Sci. 7 (6), 476.
Sarikaya, Ruhi, Hinton, Geoffrey E., Deoras, Anoop, 2014. Application of deep belief networks for natural language understanding. IEEE/ACM Trans. Audio, Speech Lang. Process. 22 (4), 778–784.
Schlosser, Joel, Chow, Christopher K., Kira, Zsolt, 2016. Fusing LIDAR and images for pedestrian detection using convolutional neural networks. In: ICRA, pp. 2198–2205.
Schmidt, Jürgen, Braunagel, Christian, Stolzmann, Wolfgang, Karrer-Gauss, Katja, 2016. Driver drowsiness and behavior detection in prolonged conditionally automated drives. In: IV, pp. 400–405.
Sermanet, Pierre, LeCun, Yann, 2011. Traffic sign recognition with multi-scale convolutional networks. In: IJCNN, pp. 2809–2813.
Shao, Jing, Loy, Chen Change, Kang, Kai, Wang, Xiaogang, 2016. Slicing convolutional neural network for crowd video understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 27-30, 2016. CVPR 2016, Las Vegas, NV, USA, pp. 5620–5628.
Silver, David, Schrittwieser, Julian, Simonyan, Karen, Antonoglou, Ioannis, Huang, Aja, Guez, Arthur, Hubert, Thomas, Baker, Lucas, Lai, Matthew, Bolton, Adrian, et al., 2017. Mastering the game of go without human knowledge. Nature 550 (7676), 354.
Sindagi, Vishwanath A., Patel, Vishal M., 2018. A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recogn. Lett. 107 (Biometrics), 3–16 Video Surveillance-oriented.
Siripanpornchana, Chaiyaphum, Panichpapiboon, Sooksan, Chaovalit, Pimwadee, 2016. Travel-time prediction with deep learning. In: TENCON. IEEE, pp. 1859–1862.
Song, Jingkuan, Zhang, Hanwang, Li, Xiangpeng, Gao, Lianli, Wang, Meng, Hong, Richang, 2018. Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Trans. Image Process. 27 (7), 3210–3221.
Srivastava, Nitish, Hinton, Geoffrey E., Krizhevsky, Alex, Sutskever, Ilya, Salakhutdinov, Ruslan, 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Sun, Fangzhou, Dubey, Abhishek, White, Jules, 2017. Dxnatłdeep neural networks for explaining non-recurring traffic congestion. In: Big Data. IEEE, pp. 2141–2150.
Sun, Jiayu, Shao, Jie, He, Chengkun, 2017. Abnormal event detection for video surveillance using deep one-class learning. Multimedia Tools Appl.
Sun, Jiayu, Wang, Xinzhou, Xiong, Naixue, Shao, Jie, 2018. Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access 6, 33353–33361.
Sutskever, Ilya, Martens, James, Hinton, Geoffrey E., 2011. Generating text with recurrent neural networks. In: ICML, pp. 1017–1024.
Sutskever, Ilya, Martens, James, Dahl, George E., Hinton, Geoffrey E., 2013. On the importance of initialization and momentum in deep learning. In: ICML, pp. 1139–1147.
Sutskever, Ilya, Vinyals, Oriol, Le, Quoc V., 2014. Sequence to sequence learning with neural networks. In: NIPS, pp. 3104–3112.
Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott E., Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, Rabinovich, Andrew, 2015. Going deeper with convolutions. In: CVPR, pp. 1–9.
Tang, Tianyu, Zhou, Shilin, Deng, Zhipeng, Lei, Lin, Zou, Huanxin, 2017. Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks. Remote Sens. 9 (11), 1170.
Tang, Tianyu, Zhou, Shilin, Deng, Zhipeng, Lei, Lin, Zou, Huanxin, 2017. Fast multidirectional vehicle detection on aerial images using region based convolutional neural networks. In: IGARSS, pp. 1844–1847.
Tayara, Hilal, Soo, Kim Gil, Chong, Kil To, 2018. Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. IEEE Access 6, 2220–2230.
Teh, Yee Whye, Hinton, Geoffrey E., 2000. Rate-coded restricted boltzmann machines for face recognition. In: NIPS, pp. 908–914.
Teney, Damien, Anderson, Peter, He, Xiaodong, van den Hengel, Anton, 2017. Tips and tricks for visual question answering: Learnings from the 2017 challenge. CoRR, abs/1708.02711.
Theis, L., Shi, W., Cunningham, A., Huszár, F., 2017. Lossy Image Compression with Compressive Autoencoders. ArXiv e-prints.
Thirumuruganathan, Saravanan, Tang, Nan, Ouzzani, Mourad, 2018. Data curation with deep learning [vision]: Towards self driving data curation. CoRR, abs/1803.01384.
Tian, Yonglong, Luo, Ping, Wang, Xiaogang, Tang, Xiaoou, 2015. Deep learning strong parts for pedestrian detection. In: ICCV, pp. 1904–1912.

Toqué, Florian, Côme, Etienne, Mahrsi, Mohamed Khalil El, Oukhellou, Latifa, 2016. Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In: ITSC. IEEE, pp. 1071–1076.
Torralba, Antonio, Fergus, Rob, Freeman, William T., 2008. 80 million tiny images: a large data set for nonparametric object and scene recognition. TPAMI 30, 1958–1970.
Tripathi, Gaurav, Singh, Kuldeep, Vishwakarma, Dinesh Kumar, 2018. Convolutional neural networks for crowd behaviour analysis: a survey. Visual Comput.
Van der Pol, Elise, Oliehoek, Frans A., 2016. Coordinated deep reinforcement learners for traffic light control.
Van Lint, J., Hoogendoorn, S., Van Zuylen, H., 2002. Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks. TRR (1811), 30–39.
Van Lint, J.W.C., Hoogendoorn, S.P., van Zuylen, Henk J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. TRC 13 (5), 347–369.
Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, Polosukhin, Illia, 2017. Attention is all you need. In: NIPS, pp. 5998–6008.
Wang, Wei, Ooi, Beng Chin, Yang, Xiaoyan, Zhang, Dongxiang, Zhuang, Yueting, 2014. Effective multi-modal retrieval based on stacked auto-encoders. PVLDB 7 (8), 649–660.
Wang, Wei, Yang, Xiaoyan, Ooi, Beng Chin, Zhang, Dongxiang, Zhuang, Yueting, 2016. Effective deep learning-based multi-modal retrieval. VLDB J. 25 (1), 79–101.
Wang, Jingyuan, Gu, Qian, Wu, Junjie, Liu, Guannan, Xiong, Zhang, 2016. Traffic speed prediction and congestion source exploration: a deep learning method. In: ICDM, pp. 499–508.
Wang, Yuan, Zhang, Dongxiang, Hu, Lu, Yang, Yang, Lee, Loo Hay, 2017. A data-driven and optimal bus scheduling model with time- dependent traffic and demand. IEEE Trans. Intell. Transp. Syst. 18 (9), 2443–2452.
Wang, Xuanhan, Gao, Lianli, Wang, Peng, Sun, Xiaoshuai, Liu, Xianglong, 2018. Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. IEEE Trans. Multimedia 20 (3), 634–644.
Wang, Lei, Zhang, Dongxiang, Gao, Lianli, Song, Jingkuan, Guo, Long, Shen, Heng Tao, 2018. Mathdqn: solving arithmetic word problems via deep reinforcement learning. In: AAAI.
Wilamowski, Bogdan M., Yu, Hao, 2010. Improved computation for levenberg-marquardt training. IEEE Trans. Neural Networks 21 (6), 930–937.
Wu, Lu-Ting, Lin, Huei-Yung, 2018. Overtaking vehicle detection techniques based on optical flow and convolutional neural network. In: VEHITS, pp. 133–140.
Wu, Yuankai, Tan, Huachun, 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. CoRR, abs/1612.01022.
Wu, Tianshu, Xie, Kunqing, Xinpin, Dong, Song, Guojie, 2012. A online boosting approach for traffic flow forecasting under abnormal conditions. In: FSKD. IEEE, pp. 2555–2559.
Wu, Yuankai, Tan, Huachun, Qin, Lingqiao, Ran, Bin, Jiang, Zhuxi, 2018. A hybrid deep learning based traffic flow prediction method and its understanding. TRC 90, 166–180.
Xue, Yadong, Li, Yicheng, 2018. A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. CACAIE 33 (8), 638–654.
Xu, Dan, Ricci, Elisa, Yan, Yan, Song, Jingkuan, Sebe, Nicu, 2015. Learning deep representations of appearance and motion for anomalous event detection. In: BMVC, pp. 8.1–8.12.
Yang, Haofan, Dillon, Tharam S., Chen, Yi-Ping Phoebe, 2017. Optimized structure of the traffic flow forecasting model with a deep learning approach. TNNLS 28 (10), 2371–2381.
Yao, Huaxiu, Wu, Fei, Ke, Jintao, Tang, Xianfeng, Jia, Yitian, Lu, Siyu, Gong, Pinghua, Ye, Jieping, Li, Zhenhui, 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In: AAAI.
Ye, Qing, Szeto, Wai Yuen, Wong, Sze Chun, 2012. Short-term traffic speed forecasting based on data recorded at irregular intervals. ITS 13 (4), 1727–1737.
Yu, Bing, Yin, Haoteng, Zhu, Zhanxing, 2017. Spatio-temporal graph convolutional neural network: a deep learning framework for traffic forecasting. CoRR, abs/1709.04875.
Yu, Rose, Li, Yaguang, Shahabi, Cyrus, Demiryurek, Ugur, Liu, Yan, 2017. Deep learning: a generic approach for extreme condition traffic forecasting. In: SIAM, pp. 777–785.
Yu, Haiyang, Wu, Zhihai, Wang, Shuqin, Wang, Yunpeng, Ma, Xiaolei, 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17 (7), 1501.
Zaklouta, Fatin, Stanciulescu, Bogdan, 2012. Real-time traffic-sign recognition using tree classifiers. ITS 13 (4), 1507–1514.
Zaklouta, Fatin, Stanciulescu, Bogdan, Hamdoun, Omar, 2011. Traffic sign classification using k-d trees and random forests. In: IJCNN, pp. 2151–2155.
Zeiler, Matthew D., Fergus, Rob, 2013. Stochastic pooling for regularization of deep convolutional neural networks. CoRR, abs/1301.3557.
Zeng, Xiaosi, Zhang, Yunlong, 2013. Development of recurrent neural network considering temporal-spatial input dynamics for freeway travel time modeling. CACAIE 28 (5), 359–371.
Zeng, Yujun, Xu, Xin, Shen, Dayong, Fang, Yuqiang, Xiao, Zhipeng, 2017. Traffic sign recognition using kernel extreme learning machines with deep perceptual features. ITS 18 (6), 1647–1653.
Zhang, Quanshi, Zhu, Song-Chun, 2018. Visual interpretability for deep learning: a survey. CoRR, abs/1802.00614.
Zhang, Lei, Yang, Fan, Zhang, Yimin Daniel, Zhu, Ying Julie, 2016. Road crack detection using deep convolutional neural network. In: ICIP, pp. 3708–3712.
Zhang, Junbo, Zheng, Yu, Qi, Dekang, 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI, pp. 1655–1661.
Zhang, Dongxiang, Guo, Long, Nie, Liqiang, Shao, Jie, Wu, Sai, Shen, Heng Tao, 2017. Targeted advertising in public transportation systems with quantitative evaluation. ACM Trans. Inf. Syst. 35 (3), 20:1–20:29.
Zhang, Jianming, Huang, Qianqian, Wu, Honglin, Liu, Yukai, 2017. A shallow network with combined pooling for fast traffic sign recognition. Information 8 (2), 45.
Zhang, Allen, Wang, Kelvin C.P., Li, Baoxian, Yang, Enhui, Dai, Xianxing, Peng, Yi, Fei, Yue, Liu, Yang, Li, Joshua Q., Chen, Cheng, 2017. Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. CACAIE 32 (10), 805–819.
Zhang, Junbo, Zheng, Yu, Qi, Dekang, Li, Ruiyuan, Yi, Xiuwen, Li, Tianrui, 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. Artif. Intell.
Zhang, Zhenhua, He, Qing, Gao, Jing, Ni, Ming, 2018. A deep learning approach for detecting traffic accidents from social media data. TRC 86, 580–596.
Zhang, Dongxiang, Li, Yuchen, Cao, Xin, Shao, Jie, Shen, Heng Tao, 2018. Augmented keyword search on spatial entity databases. VLDB J. 27 (2), 225–244.
Zhang, Shanshan, Yang, Jian, Schiele, Bernt, 2018. Occluded pedestrian detection through guided attention in cnns. In: CVPR, pp. 6995–7003.
Zhao, Zheng, Chen, Weihai, Wu, Xingming, Chen, Peter CY, Liu, Jingmeng, 2017. Lstm network: a deep learning approach for short-term traffic forecast. IET Intell. Transp. Syst. 11 (2), 68–75.
Zheng, Zuduo, 2014. Recent developments and research needs in modeling lane changing. Transport. Res. Part B: Methodol. 60, 16–32.
Zhong, Jiandan, Lei, Tao, Yao, Guangle, 2017. Robust vehicle detection in aerial images based on cascaded convolutional neural networks. Sensors 17 (12), 2720.