

Forecasting Traffic Speed during Daytime from Google Street View Images using Deep Learning

Junfeng Jiao¹  and Huihai Wang¹ 

Transportation Research Record
1–11

© National Academy of Sciences:
Transportation Research Board 2023



Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/03611981231169531

journals.sagepub.com/home/trr



Abstract

Traffic forecasting plays an important role in urban planning. Deep learning methods outperform traditional traffic flow forecasting models because of their ability to capture spatiotemporal characteristics of traffic conditions. However, these methods require high-quality historical traffic data, which can be both difficult to acquire and non-comprehensive, making it hard to predict traffic flows at the city scale. To resolve this problem, we implemented a deep learning method, SceneGCN, to forecast traffic speed at the city scale. The model involves two steps: firstly, scene features are extracted from Google Street View (GSV) images for each road segment using pretrained Resnet18 models. Then, the extracted features are entered into a graph convolutional neural network to predict traffic speed at different hours of the day. Our results show that the accuracy of the model can reach up to 86.5% and the Resnet18 model pretrained by Places365 is the best choice to extract scene features for traffic forecasting tasks. Finally, we conclude that the proposed model can predict traffic speed efficiently at the city scale and GSV images have the potential to capture information about human activities.

Keywords

data and data science, artificial intelligence, supervised learning, big data, speed data, travel time

In the process of urban expansion, populations can rapidly increase in areas disconnected from places of work, resources, and leisure, leading to problems such as traffic congestion, crashes, and air pollution. The ability to forecast traffic, therefore, is vital to the success of traffic control, traffic safety improvements, and CO₂ emission reductions. Traditionally, traffic forecasting relies on data-driven approaches, such as the autoregressive integrated moving average (ARIMA) model, linear regression models, and theory-driven traffic simulation models, such as the queuing theory model (1–6). However, these methods all assume that the traffic operates under ideal conditions, making them inefficient in a large-scale transportation system analysis with massive real-time data.

The development of machine learning provides us new approaches to capture more complex information from existing traffic datasets. These approaches—known as nonparametric methods—such as k-nearest neighbor (KNN), Bayesian network, and support vector machine (SVM) have been successfully applied in previous traffic prediction research (7–9). Recently, deep learning, an advanced type of machine learning, outperformed

traditional parametric as well as non-parametric machine learning approaches in traffic prediction accuracy (10). Non-deep-learning methods perform well in capturing the temporal dependency of traffic flow; however, they are not able to take spatial information into account, which is another vital factor which influences the accuracy of traffic forecasting.

To address this problem, graph neural networks (GNN) are integrated with deep learning models. Different kinds of recurrent neural networks (RNN), such as long short-term memory (LSTM) and gated recurrent unit (GRU), are used in these models to glean temporal information from historical traffic data. GNN are employed to integrate the spatial features of nodes in road networks. Examples include the highly performed T-GCN, DST-GCNN, STGNN, GaAN, and so on (11–14). Even though these models reach a high level of

¹Urban Information Lab, The University of Texas at Austin, Austin, TX

Corresponding Author:

Huihai Wang, hw9998@utexas.edu

accuracy for traffic flow prediction, there are two problems that still need to be solved. Firstly, these models cannot capture the physical characteristics of roads such as number of lanes and road condition. The function of surrounding areas, which cannot be learned by existing RNN and GNN models, is another significant factor that may influence the traffic flow and the relationship between connected roads. Secondly, these models are trained based on high-quality historic traffic data collected from specific roads equipped with traffic sensors, which makes it difficult to predict the traffic condition at city scale.

For these two problems, Google Street View (GSV) images have the potential to capture the road conditions and the information of functions of its surrounding area. The development of convolutional neural networks (CNN), such as VGG, ResNet, YOLO, and Mask RCNN, provide efficient approaches to extract scene features from GSV images (15, 16). These scene features include: urban functions, urban built environment, place characteristics, and so forth (17–19). Additionally, GSV images can serve as a supplement to existing historical traffic data to predict traffic flow at the city scale. With these scene features, it is possible to predict hourly traffic flows for streets without historical data.

To address the challenges of existing integrated GNN and RNN models and to prove the ability of scene features extracted from GSV images to predict traffic flow, in this paper we propose a framework to predict traffic flow hourly from GSV images by combining CNN and GNN. A CNN framework will be used to extract scene features from GSV images for each street and these features will be input to a GNN to capture the spatial information of the road network.

Literature Review

Traffic forecasting methods can be divided into two categories: parametric and nonparametric. Parametric methods construct the model structure based on certain theoretical assumptions with parameters calculated based on empirical data (20). Nonparametric methods, also known as data-driven methods, involve a more complex model structure that can be trained without prior knowledge or theoretical assumptions.

Parametric Approaches

Examples of parametric approaches for traffic forecasting include the autoregressive integrated moving average (ARIMA) model, the linear regression model, and the Kalman filter model (1–5, 21). ARIMA—also written as ARIMA(b, d, q)—is one of the most widely used models to forecast traffic flow. The autoregressive, integrated, and moving average polynomial orders are essential parameters used to build the ARIMA time series model

(22, 23). However, since traffic conditions are neither stationary nor linear, parametric approaches are not applicable to the rapid changes in traffic flow.

Nonparametric Methods

Nonparametric approaches such as KNN, Bayesian network, SVM, and artificial neural networks can successfully capture the complex and nonlinear characteristics of traffic flow (8, 24). The KNN method requires a high-quality traffic flow database. This method searches for data that are similar to observed data at certain locations, such as a station, then similar traffic flow series are used to forecast the traffic flow of the station (7, 25). Zhang et al. present a KNN model to predict urban expressway flow with up to 90% accuracy (26). The SVM method involves mapping data to a high-dimensional feature space and performing linear regressions within that space (27). Ling et al. introduced a multi-kernel SVM (MSVM) to predict traffic flow, and a novel adaptive particle swarm optimization (APSO) algorithm to optimize the parameters of MSVM. Their results show that this algorithm can make timely and adaptive predictions during peak hour when the traffic conditions change rapidly (28).

Nonparametric models can also be integrated with other nonparametric models to improve overall performance. Ahn et al. used both support vector regression and Bayesian classifiers to conduct real-time traffic flow prediction (29). Random forest and support vector regression are integrated to perform short-term traffic flow forecasting (30). Additionally, the combination of Kalman filtering and KNN, KNN and SVM, KNN and LSTM, KNN and neural networks, and so forth, are performed to predict short-term traffic flow (31–34).

With the rapid development of deep learning, deep neural network models are now widely used to forecast traffic flow (35–41). At present, modeling methods based on deep neural networks achieve the most accurate results because of their ability to extract dynamic traffic features. These methods can identify traffic features without prior knowledge or assumptions, handle multi-dimensional data and flexible model structures, and employ strong generalization and learning abilities (11, 20, 35). In recent years, RNNs and their variants, LSTM and GRU, have received attention because of their self-circulation mechanism, which allows them to learn temporal dependence, and their ability to outperform other types of deep neural networks (11). Tian and Pan proposed an LSTM model to determine optimal time lags dynamically. Their results show that LSTM achieves higher accuracy than other methods including random walk (RW), SVM, single-layer feed forward neural network (FFNN) and stacked autoencoder (SAE) (42). However, these models only focus on temporal characteristics, meaning they cannot capture the spatial dependencies of traffic flow. GNN

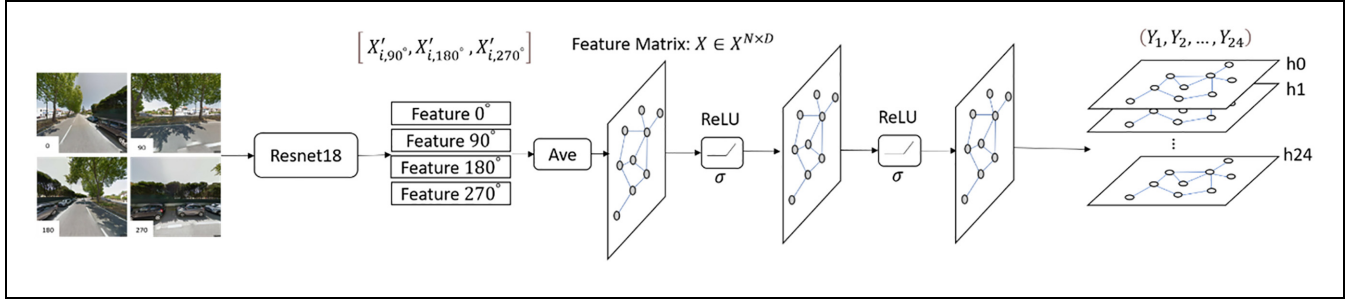


Figure 1. Structure of scene graph convolutional network (GCN).

can aggregate and transform traffic information through edges in road networks (43–45). This allows GNN to capture the spatial dependencies of traffic flow, which improves the accuracy of traffic forecasting. More and more researchers have combined these deep learning models with GNN to capture the spatiotemporal characteristics of traffic flow. Wu et al. proposed a graph attention LSTM network (GAN-LSTM) to capture spatiotemporal correlations for traffic flow forecasting, and found GAN-LSTM outperformed other multi-link traffic flow forecasting models: diffusion convolutional recurrent neural network (DCRNN), LSTM, and feed forward neural network (FNN)(46). A gated CNN can be combined with a graph convolutional network (GCN) to form a spatiotemporal GCN (STGCN), which captures comprehensive spatiotemporal correlations and runs much faster with fewer parameters (13). Zhang et al. proposed a model based on gated attention networks (GaAN) to extract spatiotemporal characteristics of traffic flow (14). The combination of RNN and GNN requires both a historical dataset to train the model and observation traffic flow data to forecast traffic conditions. CNN and RNN can also be combined to capture the spatiotemporal characteristics of traffic flow at the city scale (47, 48). This method involves transforming the road network to raster maps, with the values of each cell representing the condition of traffic, then typical RNN networks are used to capture the temporal characteristics of traffic data for each cell. With these models, we can learn the spatiotemporal features of traffic flow data at the city level; however, these models require traffic data on all the streets, such as trajectory data, which limits this approach to situations where a city-wide dataset is accessible. The need for both historical and observation data makes these models difficult to apply to the entire city.

Methodology

Problem Definition

In the problem of traffic forecasting, we intend to predict hourly traffic speeds during workdays based on scene

features extracted from GSV images. Specifically, the scene features of each road are represented on a traffic network model. We describe the traffic network as an unweighted graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_N\}$ is a set of road nodes (we treat each road as a node), N is the number of the nodes, and E is a set of edges representing the connection of each road. The adjacency matrix, $A \in \{R^{N \times N}\}$, can also represent the connection between roads. For each element in the adjacency matrix, A_{ij} , which represents the connection between node i and node j , there are only two possible values (0 means no connection; 1 means they are connected). The scene features extracted from GSV images for road nodes can be represented as a feature matrix, $X \in X^{N \times D}$, in which $X_i = [x_1, x_2, \dots, x_D]$ represents features of node i , where D represents the dimension of scene features for each node.

The hourly traffic speed information for each node can be represented as a sequence $y = (Y_1, Y_2, \dots, Y_{24})$ on the traffic network, where each Y_t is the traffic speed at time t .

Thus, the problem of traffic speed forecasting can be defined as building a model f on the traffic network G and feature matrix $X_i = [x_1, x_2, \dots, x_D]$ to predict the hourly traffic speed $y = (Y_1, Y_2, \dots, Y_{24})$, which is shown as (1):

$$(Y_1, Y_2, \dots, Y_{24}) = f(G; [x_1, x_2, \dots, x_D]) \quad (1)$$

where

D = the dimension of extracted scene features for each street node.

Method

The proposed scene graph CNN framework consists of two parts: the CNN and the GNN. As shown in Figure 1, we first use a pretrained CNN model, Resnet18, to extract scene features from GSV images for each street node, then the average of these features is calculated to construct the feature matrix X . Secondly, the generated scene features are entered into a GCN to capture the spatial dependency.

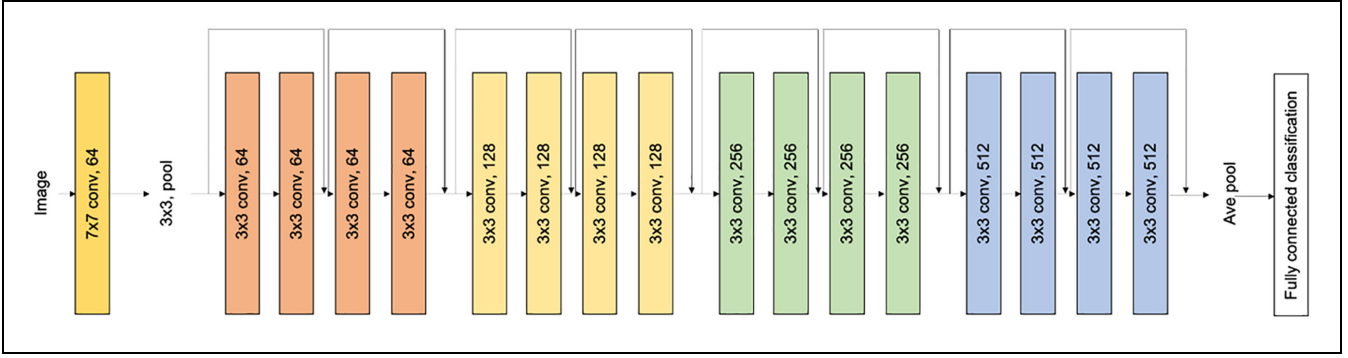


Figure 2. Structure of Resnet18.

Finally, we get a predicted hourly traffic speed sequence, $\hat{y} = (\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_{24})$, for each road node, where each \hat{Y}_t represents the traffic speed at time t .

Scene Features Extraction. Four GSV images at heading angles of 0° , 90° , 180° , and 270° for each street node are generated using the GSV static API. For each node, we use Resnet18 to extract scene features from GSV. Features in all heading angles are generated, $X'_i = [X'_{i,90^\circ}, X'_{i,180^\circ}, X'_{i,270^\circ}]$, where X'_i is the list of features at each of the four heading angles of node i . Then, the average value of each location is calculated, $X_{i,j} = \frac{1}{4}(X'_{i,0^\circ,j} + X'_{i,90^\circ,j} + X'_{i,180^\circ,j} + X'_{i,270^\circ,j})$, where $X_{i,j}$ is the feature of node i at dimension j , and $i \in N, j \in D$. We use a pretrained Resnet18 to extract scene features. The model is trained by the Places365 dataset with a high top-5 accuracy of 85.07% (49). It employs an identity shortcut connection mechanism to skip one or more layers when training the model, so that more layers could be included in the model resulting in higher accuracy. Resnet18 requires relatively fewer parameters and less computational cost to extract features from images than other methods. The structure of our Resnet is shown as Figure 2.

Spatial Dependence Modeling. The traffic speed in a street node is influenced not only by its own features, but also by the other street nodes connected to it. GCN has been successfully used in extracting spatial dependencies of traffic networks to predict traffic flow with high accuracy. The mechanism of GCN is to construct a filter in the Fourier domain based on adjacency matrix A and feature matrix X . For each street node, the filter will aggregate the features of its connected nodes, then several convolutional layers are stacked to extract further and more complex spatial features. The GCN can be expressed as:

$$X^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^l W^l) \quad (2)$$

where

$\tilde{A} = A + I_N$ = the adjacent matrix with self-connections,

I_N = the identity matrix,

$\tilde{D} = \sum_j \tilde{A}_{ij}$, \tilde{D} = the degree matrix,

X^l = the output feature matrix of layer l ,

W^l = the trainable weights of layer l , and

$\sigma(\cdot)$ = the activation function (ReLU).

In this research, we adopt a two-layer GCN to extract spatial features from the traffic network. The process can be expressed as:

$$\hat{y} = \tilde{A} \sigma(\tilde{A} X_0 W_0) W_1 \quad (3)$$

where

$\hat{y} \in R^{N \times 24}$ = the predicted hourly traffic speed,

$\tilde{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, $X_0 \in R^{N \times D}$ = the scene features extracted from GSV, and

$W_0 \in R^{D \times T}$ = trainable weight matrix from the hidden to the output layer, and

$W_1 \in R^{T \times 24}$ = trainable weight matrix from the hidden to the output layer.

Experiments

In this section, we set up experiments to evaluate the performance of our proposed SceneGCN framework. We first introduce the dataset used for training our model and evaluation metrics. Then, different CNN models are employed to extract scene features. Finally, we present the experiment results and interpretation of our proposed model.

Data Description

Taxi Trajectory Dataset. The performance of the proposed SceneGCN model is evaluated on a taxi trajectory dataset collected in the city of Porto, Portugal (50). A total of

442 taxis were equipped with mobile data terminals to collect trajectory data for a complete year (from July 1, 2013, to June 30, 2014). In this dataset, GPS signals are collected every 15 s and timestamps of the start point for each trip are recorded. We select Monday to Friday (except holidays or other special days and days before these days) to prepare our training and testing dataset.

Google Street View Imagery (GSV). GSV provides us 360° panoramic scenes along streets at the same viewing angles as pedestrians. It provides a huge amount of imagery to explore urban built environments. The Google Cloud platform provides us a GSV static API to download these images automatically. The necessary parameters include location (latitude and longitude of the images), pano-ID (the specific panorama ID), output size of each image, and key of the API, as well as other optional parameters, such as the heading (compass heading of the camera), fov (horizontal field of view of the image) and pitch (the up and down angle of the camera relative to the GSV vehicle). In this research, we download GSV images for each road segment at the central point of each road at heading angles 0°, 90°, 180°, and 270°. Finally, 18,324 images are downloaded to gather scene features for each road segment. These images are used to extract scene features, which are possibilities of different scene types, such as apartments, highways, bridges, parking lots, and so forth. Since these features are extracted from a pretrained Resnet model with a high accuracy, the time the images are captured will not influence the extraction of the feature (49). To match the time of the images and taxi dataset, we downloaded the GSV images from year 2013 to 2014. Figure 3 shows examples of our downloaded images.

Evaluation Metrics

To evaluate the performance of the SceneGCN, we use three metrics:

- 1) Mean absolute error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - \hat{f}_i| \quad (4)$$

- 2) Root mean squared error (RMSE):

$$RMSE = \left[\frac{1}{n} \sum_{i=1}^n (|f_i - \hat{f}_i|)^2 \right]^{\frac{1}{2}} \quad (5)$$

- 3) Mean absolute percentage error (MAPE):

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|f_i - \hat{f}_i|}{f_i} \quad (6)$$



Figure 3. Examples of downloaded Google Street View at heading angles of 0° (top left), 90° (top right), 180° (bottom left), and 270° (bottom right).

where

f_i = the real traffic speed, and
 \hat{f}_i = the predicted traffic speed.

Model Parameters

Our proposed model deals with two tasks: scene feature extraction from GSV using CNN, and traffic speed prediction based on GCN. For scene feature extraction, we adopt two Resnet18 models pretrained by two different datasets:

Places365: The Places365 dataset classifies images into 365 categories of scenes, such as highway, forest, field, street, church, plaza, and so on (51). The weights of the model are trained by 18 million images and predicts the category of each image with high accuracy (49).

ImageNet: The ImageNet dataset classifies images into 1,000 object categories, including different kinds of cars, animals, vegetation, infrastructure, and so on.

Places365 aims to recognize scene features at a larger, more general scale, while ImageNet focuses on specific objects. We compared the performance of these two pretrained models on traffic speed forecasting to determine the most suitable prediction model. In each pretrained model, we keep the pretrained weights, and test whether containing the classification layer will influence the performance of our model.

Table 1. Evaluation Matrix

Model		MAPE	RMSE	MAE
Places365	365 scenes	0.348	15.91	14.55
	512 features	0.347	15.87	14.47
ImageNet	1,000 classes	0.377	17.02	15.64
	512 features	0.375	17.02	15.67

Note: MAE = mean absolute error; MAPE = mean absolute percentage error; RMSE = root mean squared error.

In total, four experiments are conducted to evaluate the performance of our proposed model: Resnet18 pretrained by Places365 containing the classification layer, which generates a 365 dimension set of features for each road node; Resnet18 pretrained by Places365 without classification layer, which generates a 512 dimension set of features for each road node; Resnet18 pretrained by ImageNet containing the last classification layer and without the last classification layer, which obtains a 1,000 dimension set and 512 dimension set of features for each road node, respectively. For each experiment, the generated features are entered into a two-layer GCN to predict traffic speed. Finally, we use the MAE as the loss function (L1Loss) to reduce the error between predicted speed and real speed during the training process. The loss function is defined as:

$$loss = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (7)$$

where

- n = the number of predicted values,
- \hat{y}_i = the predicted traffic speed value, and
- y_i = the real traffic speed value.

Experimental Results

We first processed the taxi trajectory dataset, then we obtained the training dataset, validation dataset, and test dataset with sizes of 600, 300, and 303, respectively. Scene features were extracted and entered into the GCN to train the model. The most accurate result for each experiment was generated after training about 2,500 epochs. We used the test dataset to evaluate our model. The performance of our models is shown in Table 1. From the evaluation matrix we find that the GCN trained by features extracted by the Places365 pretrained model has MAPE, RMSE, and MAE of around 0.35, 15.9, and 14.5 respectively, which are lower than that of the ImageNet pretrained models, which are around 0.37, 17.02, and 15.6, respectively. The classification layer has little effect on performance of our proposed model. A total of 365 scenes features obtained by Resnet18 with classification layer predict the traffic speed at similar errors as 512 scenes

features extracted by Resnet18 without classification layer. Traffic speeds predicted by 365 scene features are more stable than that predicted by 512 features. So, we chose the 363 scene features extracted by Resnet18 (with classification layer, and pretrained by Places365) as our feature matrix to train the GCN model.

Figure 4 shows a selection of the results of our test dataset. We can see that our proposed model can predict traffic speed patterns during the workday with high accuracy. The predicted results in these figures show that, overall, traffic speeds in the city of Porto experience two rapid declines from 6 a.m. to 9 a.m. and 4 p.m. to 7 p.m. The lowest speeds occur at 8 a.m. and between 6 p.m. and 7 pm. After these valley hours, traffic speeds will recover to normal speed within one or two hours. The traffic speeds at night (from 10 p.m. to 5 a.m.) are faster than during daytime. The model can predict traffic speeds ranging from 0 km/h to 100 km/h with a high degree of accuracy. However, some road nodes are overestimated or underestimated. Table 2 shows that, in our test dataset, 17 road nodes are overestimated, 24 road nodes are underestimated, and 262 road nodes are predicted correctly. More detailed reasons on prediction errors are analyzed in the following paragraph.

Figure 5 shows the result map of our test dataset. Purple lines are road nodes that are underestimated, red lines are road nodes that are overestimated, and green lines are road nodes that are predicted correctly. From this map we can identify that the road nodes that are underestimated are mainly located at the entrance and exit of highways. This is because these road nodes connect high-speed roads to roads with low speeds, which will be aggregated and result in an underestimated speed prediction through GCN. Also, the speeds at the entrances and exits of highways are usually lower than those on normal highway nodes. For overestimated road nodes, they are mainly located near highways. This is because some highway nodes capture features of surrounding buildings, which can cause errors that confuse these nodes with other streets with similar building scenes. Also, some overestimated roads are connected to highways, therefore information from these highways will be transferred to these overestimated roads through the GCN.

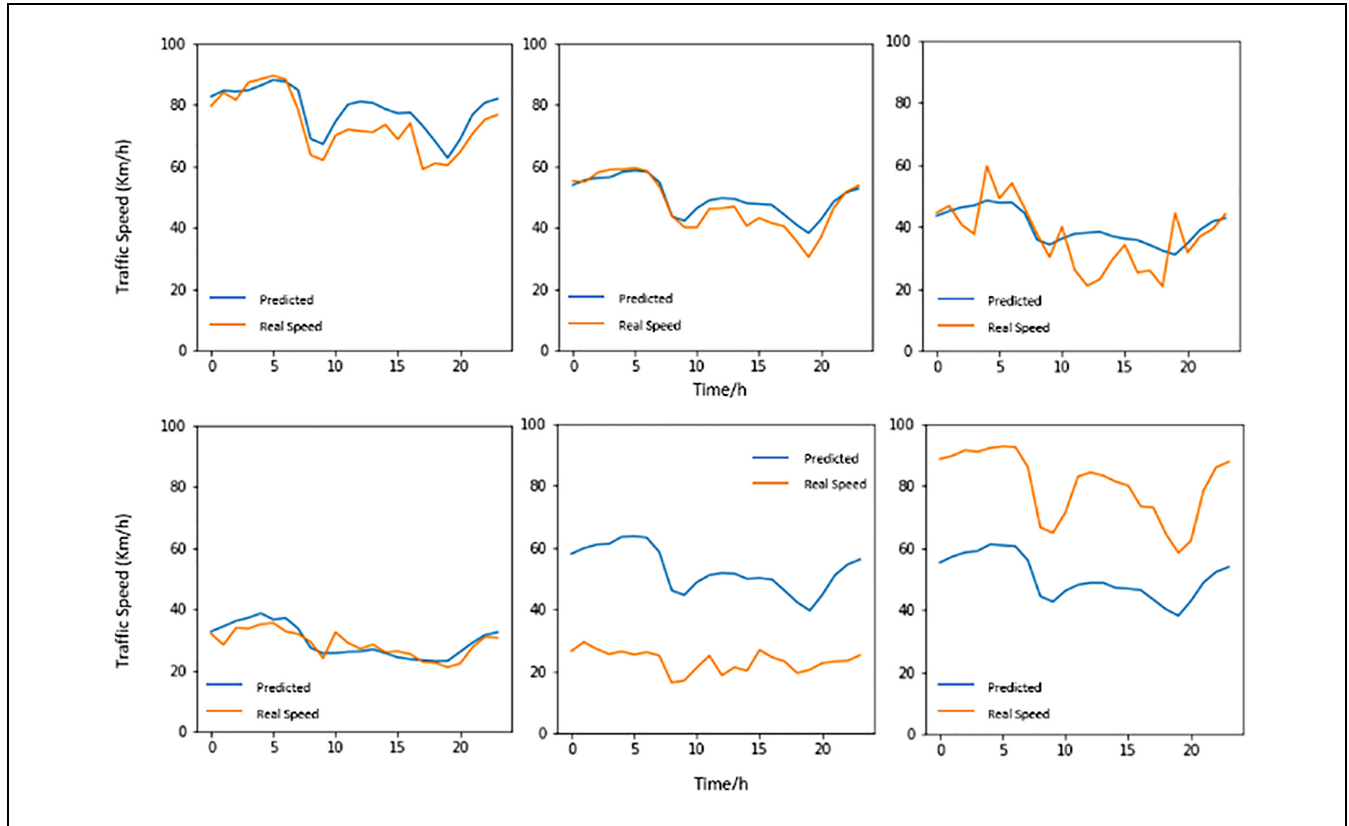


Figure 4. Result of test data.

Note: 'top left' = the figure at the first row, the first column; 'top middle' = the figure at the first row, the second column; 'bottom left' = the figure at the second row, the first column; 'bottom middle' = the figure at the second row, the second column; 'bottom right' = the figure at the second row, the third column.

Table 2. Accuracy of SceneGCN

	Count	Percentage
Overestimate	17	5.6%
Underestimate	24	7.9%
Correct	262	86.5%

Finally, we forecasted traffic speeds for each hour of the day for the city of Porto. Figure 6 shows predicted traffic speeds at the city scale (taken at 6 a.m., noon, 6 p.m., and midnight.). The map shows that, spatially, the traffic speed of highways is much higher than that of city roads. Traffic speed is lowest at the center of the city. Temporally, traffic speed is higher at night than at any other time, and traffic speed is lowest during commuting hours (6 p.m. in this map). The red lines in Figure 6 have the lowest traffic speed, revealing areas of traffic congestion.

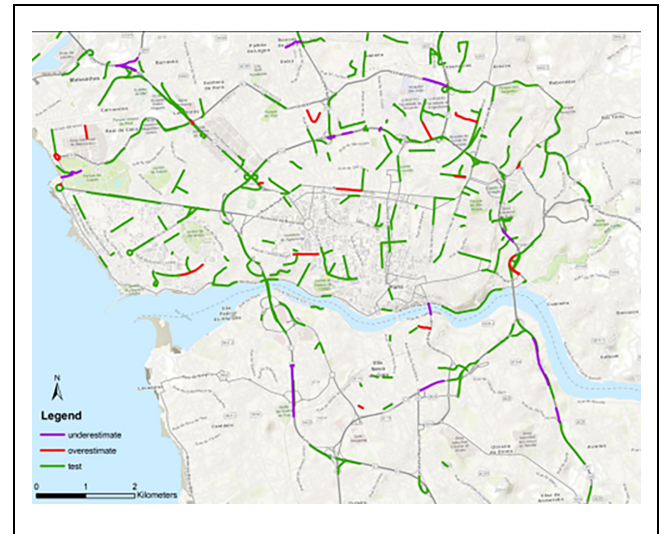


Figure 5. Result map of test dataset.

Note: Green lines = road nodes that are predicted accurately; purple lines = road nodes that are underestimated; red lines = road nodes that are overestimated.

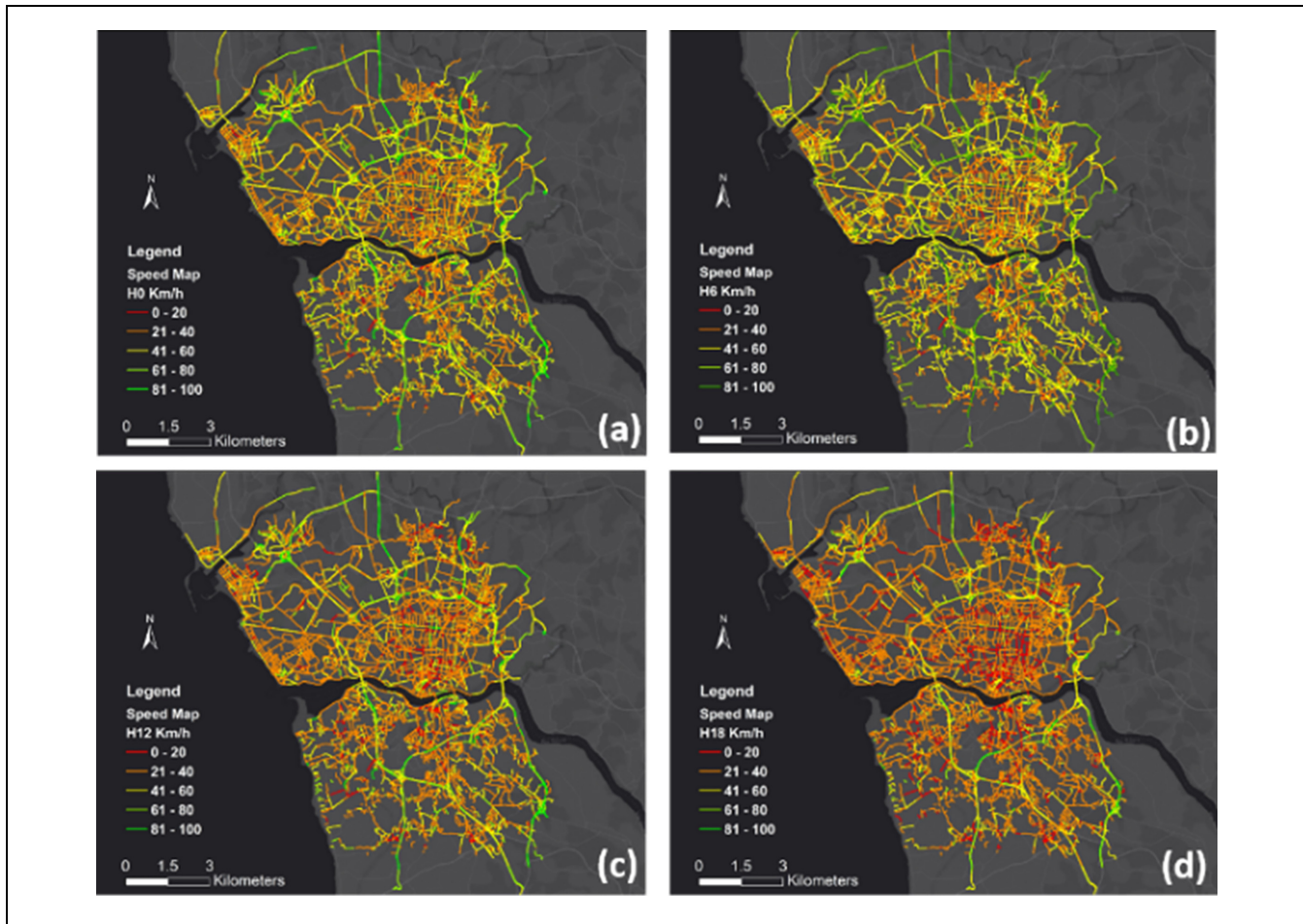


Figure 6. Predicted traffic speed of Porto: (a) traffic speed at midnight, (b) traffic speed at 6 a.m., (c) traffic speed at noon, (d) traffic speed at 6 p.m.

Conclusion

As populations in urban areas increase rapidly, traffic forecasting becomes more and more important. Existing methods rely on historical traffic data, which makes it hard to forecast traffic at city scale. In this research, we proposed a method based on SceneGCN that uses scene features extracted from GSV to predict traffic speeds for each hour of the day. Different scene features extracted from pretrained Resnet18 models are compared and a GCN model is used to predict traffic speed for each hour of the day. The proposed model has the ability to predict traffic speed at city scale, which can serve as a supplement to traffic flow datasets which are usually collected at limited sites. Our method also illustrated the potential of scene features in traffic flow prediction. Further research can be conducted on the basis of this paper, for example, to improve traffic flow prediction by integrating scene features to existing spatiotemporal models. According to the results, we have two conclusions. First, our proposed

model can capture the spatiotemporal characteristics of traffic flow. From the forecasted traffic speed map, we find, as suspected, that areas with more human activities, such as the central city, have lower traffic speeds. Also, as suspected, traffic speed changes during commuting hours, and traffic congestion may occur during these hours. Secondly, scene features extracted from GSV images have the potential to be used to estimate the built environment and urban functions. Traffic speed is influenced by human activities and road conditions, and the good performance of our model indicates that these factors can be reflected in the GSV images.

Our proposed model has some limitations. First, we adopt GCN to capture the spatial characteristics of the traffic network. It requires a high-quality traffic network map and any changes to the network may require retraining the model. Second, the quality of GSV images influences the accuracy of our model significantly. Some highways capture features of surrounding buildings, which may cause errors in streets with similar building

scenes. Third, even though GSV images can reflect human activities and road conditions to some degree, other characteristics may influence traffic speed, such as the posted speed limitation and density of population, as well as the availability of public transportation. In the future, we plan to adopt more social-economic factors and urban points of interest (POI) data to our model to improve the forecasting accuracy and improve the temporal resolution of our model.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: H. Wang; data collection: H. Wang; analysis and interpretation of results: H. Wang; draft manuscript preparation: H. Wang, J. Jiao. All authors reviewed the results and approved the final version of the manuscript.


Declaration of Conflicting Interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research was supported by Good System, a research grand challenge at the University of Texas at Austin and was supported by the National Science Foundation (NSF), ART-AI: Convergent, Responsible, and Ethical Artificial Intelligence Training Experience for Robotics, and National Science Foundation (NSF), SCC-CIVIC-PG Track A: Community Hub for Smart Mobility. UIL research was supported by the NSF Grants (2043060, 2133302, 1952193, 2125858, 2236305) USDOT consortium of Cooperative Mobility for Competitive Megaregions, Good System at the University of Texas at Austin and The Mitre Corporation.

ORCID iDs

Junfeng Jiao  <https://orcid.org/0000-0002-7272-8805>

Huihai Wang  <https://orcid.org/0000-0002-7343-3726>

References

1. Levin, M., and Y. D. Tsao. On Forecasting Freeway Occupancies and Volumes (abridgment). *Transportation Research Record*, 1980. 773: 47–49.
2. Han, C., S. Song, and C. H. Wang. A Real-Time Short-Term Traffic Flow Adaptive Forecasting Method Based on ARIMA Model. *Acta Simulata Systematica Sinica*, Vol. 7, No. 4, 2004, p. 3.
3. Van Der Voort, M., M. Dougherty, and S. Watson. Combining Kohonen Maps with Arima Time Series Models to Forecast Traffic Flow. *Transportation Research Part C, Emerging Technologies*, Vol. 4, No. 5, 1996, pp. 307–318.
4. Dudek, G. Pattern-Based Local Linear Regression Models for Short-Term Load Forecasting. *Electric Power Systems Research*, Vol. 130, 2016, pp. 139–147.
5. Sun, H., C. Zhang, and B. Ran. Interval Prediction for Traffic Time Series using Local Linear Predictor. *The 7th International IEEE Conference on Intelligent Transportation Systems*, Washington, WA, October 3–6, 2004, IEEE, New York, NY, p. 410–415.
6. Heidemann, D. A Queueing Theory Model of Nonstationary Traffic Flow. *Transportation Science*, Vol. 35, No. 4, 2001, pp. 405–412.
7. Davis, G. A., and N. L. Nihan. Nonparametric Regression and Short-Term Freeway Traffic Forecasting. *Journal of Transportation Engineering*, Vol. 117, No. 2, 1991, pp. 178–188.
8. Sun, S., C. Zhang, and G. Yu. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 1, 2006, pp. 124–132.
9. Deshpande, M., and P. R. Bajaj. Performance Analysis of Support Vector Machine for Traffic Flow Prediction. *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICCC)*, Jalgaon, India, December 22–24, 2016, IEEE, New York, NY, pp. 126–129.
10. Yang, H., C. Liu, M. Zhu, X. Ban, and Y. Wang. How Fast Will You Drive? Predicting Speed of Customized Paths by Deep Neural Network. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 3, 2022, pp. 2045–2055.
11. Zhao, L., Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, No. 9, 2020, pp. 3848–3858.
12. Chen, K., F. Chen, B. Lai, Z. Jin, Y. Liu, K. Li, L. Wei, et al. Dynamic Spatio-temporal Graph-based CNNs for Traffic Prediction. *ArXiv181202019 Cs* [Internet]. 2020. <http://arxiv.org/abs/1812.02019>. Accessed April 3, 2021.
13. Yu, B., H. Yin, and Z. Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, AAAI Press, Palo Alto, California, 2018, pp. 3634–3640.
14. Zhang, J., X. Shi, J. Xie, H. Ma, I. King, and D. Y. Yeung. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. *ArXiv180307294 Cs* [Internet]. 2018. <http://arxiv.org/abs/1803.07294>. Accessed April 3, 2021.
15. Liu, S., and W. Deng. Very Deep Convolutional Neural Network Based Image Classification Using Small Training Sample Size. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, November 3–6, 2015, IEEE, New York, NY, pp. 730–734.
16. He, K., X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv151203385 Cs* [Internet]. 2015. <http://arxiv.org/abs/1512.03385>. Accessed April 4, 2021.

17. Ye, C., F. Zhang, L. Mu, Y. Gao, and Y. Liu. Urban Function Recognition by Integrating Social Media and Street-Level Imagery. *Environment and Planning B: Urban Analytics and City Science*, Vol. 48, No. 6, 2020, pp. 1430–1444.
18. Qin, K., Y. Xu, C. Kang, and M. Kwan. A Graph Convolutional Network Model for Evaluating Potential Congestion Spots Based on Local Urban Built Environments. *Transactions in GIS*, Vol. 24, No. 5, 2020, pp. 1382–1401.
19. Zhu, D., F. Zhang, S. Wang, Y. Wang, X. Cheng, Z. Huang, and Y. Liu. Understanding Place Characteristics in Geographic Contexts through Graph Convolutional Neural Networks. *Annals of American Association of Geographers*, Vol. 110, No. 2, 2020, pp. 408–420.
20. Ma, X., Z. Tao, Y. Wang, H. Yu, and Y. Wang. Long Short-Term Memory Neural Network for Traffic Speed Prediction using Remote Microwave Sensor Data. *Transportation Research Part C: Emerging Technologies*, Vol. 54, 2015, pp. 187–197.
21. Xu, D. w., Y. d. Wang, L. m. Jia, Y. Qin, and H. h. Dong. Real-Time Road Traffic State Prediction Based on ARIMA and Kalman Filter. *Frontiers of Information Technology & Electronic Engineering*, Vol. 18, No. 2, 2017, pp. 287–302.
22. Smith, B. L., B. M. Williams, and O. R. Keith. Comparison of Parametric and Nonparametric Models for Traffic Flow Forecasting. *Transportation Research Part C: Emerging Technologies*, Vol. 10, No. 4, 2002, pp. 303–321.
23. Han, C., and S. Song. A Review of Some Main Models for Traffic Flow Forecasting. *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, Shanghai, China, October 12–15, 2003, IEEE, New York, NY, pp. 216–219.
24. Zhong, J. t., and S. Ling. Key Factors of K-Nearest Neighbours Nonparametric Regression in Short-Time Traffic Flow Forecasting. In: *Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014* (E. Qi, J. Shen, and R. Dou, eds.). Atlantis Press, Paris, 2015, pp. 9–12.
25. Zhang, T., L. Hu, Z. Liu, and Y. Zhang. Nonparametric Regression for the Short-Term Traffic Flow Forecasting. *2010 International Conference on Mechanic Automation and Control Engineering*, Wuhan, June 26–28, 2010, IEEE, New York, NY, pp. 2850–2853.
26. Zhang, L., Q. Liu, W. Yang, N. Wei, and D. Dong. An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction. *Social and Behavioral Sciences*, Vol. 96, 2013, pp. 653–662. <https://reader.elsevier.com/reader/sd/pii/S1877042813022027?token=3E46024D5B2BE47AA4850C0B4E2B6A0853BCB18EBFA346C7AADDFFB3E465D6753076074E6B4681D6463C8B1EE3A57971&originRegion=us-east-1&originCreation=20210404191820>. Accessed April 4, 2021.
27. Smola, A. J., and B. Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, Vol. 14, No. 3, 2004, pp. 199–222.
28. Ling, X., X. Feng, Z. Chen, Y. Xu, and H. Zheng. Short-Term Traffic Flow Prediction with Optimized Multi-Kernel Support Vector Machine. *2017 IEEE Congress on Evolutionary Computation (CEC)*, San Sebastian, June 5–8, 2017, IEEE, New York, NY, pp. 294–300.
29. Ahn, J., E. Ko, and E. Y. Kim. Highway Traffic Flow Prediction Using Support Vector Regression and Bayesian Classifier. In: *2016 International Conference on Big Data and Smart Computing (BigComp)*, Hong Kong, China, January 18–20, 2016, IEEE, New York, NY, pp. 239–244.
30. Zhang, L., N. R. Alharbe, G. Luo, Z. Yao, and Y. Li. A Hybrid Forecasting Framework Based on Support Vector Regression with a Modified Genetic Algorithm and a Random Forest for Traffic Flow Prediction. *Tsinghua Science and Technology*, Vol. 23, No. 4, 2018, pp. 479–492.
31. Liang, S., M. Ma, S. He, and H. Zhang. Short-Term Passenger Flow Prediction in Urban Public Transport: Kalman Filtering Combined K-Nearest Neighbor Approach. *IEEE Access*, Vol. 7, 2019, pp. 120937–120949.
32. Liu, Z., W. Du, D. m. Yan, G. Chai, and J. h. Guo. Short-Term Traffic Flow Forecasting Based on Combination of K-Nearest Neighbor and Support Vector Regression. *Journal of Highway and Transportation Research and Development*, Vol. 12, No. 1, 2018, pp. 89–96.
33. Luo, X., D. Li, Y. Yang, and S. Zhang. Spatiotemporal Traffic Flow Prediction with KNN and LSTM. *Journal of Advanced Transportation*, Vol. 2019, 2019, pp. 1–10.
34. Liu, Z., J. Guo, J. Cao, Y. Wei, and W. Huang. A Hybrid Short-term Traffic Flow Forecasting Method Based on Neural Networks Combined with K-Nearest Neighbor. *Promet – Traffic Transportation*, Vol. 30, No. 4, 2018, pp. 445–456.
35. Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Y. Wang. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, 2014, pp. 865–873.
36. Yang, H., T. S. Dillon, and Y. P. Chen. Optimized Structure of the Traffic Flow Forecasting Model with a Deep Learning Approach. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 10, 2017, pp. 2371–2381.
37. Koesdwiady, A., R. Soua, and F. Karray. Improving Traffic Flow Prediction with Weather Information in Connected Cars: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology*, Vol. 65, No. 12, 2016, pp. 9508–9517.
38. Yi, H., H. Jung, and S. Bae. Deep Neural Networks for traffic flow prediction. *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, February 13–16, 2017, IEEE, New York, NY, pp. 328–331.
39. Polson, N. G., and V. O. Sokolov. Deep Learning for Short-Term Traffic Flow Prediction. *Transportation Research Part C: Emerging Technologies*, Vol. 79, 2017, pp. 1–17.
40. Huang, W., G. Song, H. Hong, and K. Xie. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 5, 2014, pp. 2191–2201.
41. Çetiner, B., M. Sari, and O. Borat. A Neural Network Based Traffic-Flow Prediction Model. *Mathematical and Computational Applications*, Vol. 15, No. 2, 2010, pp. 269–278.

42. Tian, Y., and L. Pan. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). Chengdu, China, December 19–21, 2015, New York, NY, pp. 153–158.
43. Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, Vol. 20, No. 1, 2009, pp. 61–80.
44. Wang, X, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, et al. Traffic Flow Prediction via Spatial Temporal Graph Neural Network. In: *Proceedings of The Web Conference 2020* [Internet]. Taipei Taiwan: ACM; 2020 [cited 2021 Apr 3]. p. 1082–1092. Available from: <https://dl.acm.org/doi/10.1145/3366423.3380186>
45. Zhou, F., Q. Yang, T. Zhong, D. Chen, and N. Zhang. Variational Graph Neural Networks for Road Traffic Prediction in Intelligent Transportation Systems. *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 4, 2021, pp. 2802–2812.
46. Wu, T., F. Chen, and Y. Wan. Graph Attention LSTM Network: A New Model for Traffic Flow Forecasting. *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, Zhengzhou, China, July 20–22, 2018, IEEE, New York, NY, p. 241–245.
47. Sun, S., H. Wu, and L. Xiang. City-Wide Traffic Flow Forecasting Using a Deep Convolutional Neural Network. *Sensors*, Vol. 20, No. 2, 2020, p. 421.
48. Jia, T., and P. Yan. Predicting Citywide Road Traffic Flow Using Deep Spatiotemporal Neural Networks. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 5, 2020, pp. 3101–3111.
49. Zhou, B., A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 6, 2018, pp. 1452–1464.
50. Kaggle. *Taxi Trajectory Data*, 2021. <https://kaggle.com/craita/taxi-trajectory>. Accessed April 25, 2021.
51. *Places2: A Large-Scale Database for Scene Understanding*, 2021. <http://places2.csail.mit.edu/download.html>. Accessed April 25, 2021.