Learning Traffic Deeply as Images with Partially Observed Data

-Travel Time Prediction on I-80 E from Davis to West Sacramento

Jin Guo

1 Introduction

Predicting travel time is a critical issue in traffic modeling, which helps road users better choose their travel modes and plan routes(X. Zhang and Rice 2003). Conceptually, estimation and prediction of travel time are different in the time slots and the frameworks they used to give the hypothesis(Mori et al. 2015). Estimation algorithms calculate travel times of vehicles based on their historic trajectories. While in prediction problems, a time variable is included and the objective is to use the current and past data to forecast the travel time in future time intervals(Lindveld et al. 2007).

Methodologies used in travel time prediction include empirical models, simulation-based models, data-driven models and hybrid analysis of them. Empirical models assume traffic characteristics remain similar in the same time intervals of different days and use linear combinations of historical data to predict the future(Tenenboim and Shiftan 2018). There are three levels of traffic simulations – macroscopic, microscopic and mesoscopic models. Macroscopic models focuses on aggregate features based on mathematical equations, mean speed, flow and density of road segments are mostly used variables when simulating the evolution of traffic flows(Helbing et al. 2002). Microscopic models simulate behaviors of individual vehicles with known or estimated Origin-Destination(O-D) matrices. These models have pre-determined car following and lane changing models, which are generated from differential equations in continuous time domain or difference equations in discrete time domain(H. M. Zhang 2002). Mesoscopic simulations are compromises of macroscopic and microscopic ones, which simulate individual vehicles while use features calculated at road or network level(Burghout, Koutsopoulos, and Andréasson 2007).

Travel time prediction, like other Multiple In Single Output (MISO) systems, can be simplified to a regression problem, which has a potential solution given by parametric or non-parametric data driven models(van Lint 2004). Classic parametric models used in travel time prediction include Linear Regression, where input variables are traffic information observed by sensors and relative features like weather and time-of-day(Kwon, Coifman, and Bickel 2007), Bayesian Nets(Fei, Lu, and Liu 2011) and Time Series Analysis(Hofleitner, Herring, and Bayen 2012). With the rapid development of machine learning algorithms, non-parametric models have been widely applied to predict traffic information, such as ARIMA (Jiann-Shiou Yang 2005), KNN(Davis and Nihan 2007), SVM(C. H. Wu et al. 2003) and Decision Trees(Oh, Ritchie, and Oh 2007).

The proposal of backpropagation algorithm makes Artificial Neural Networks (ANN) and Deep Neural Network (DNN) powerful tools in predicting travel time due to their capability to handle high-dimensional data, good generalizability and portability between different data sets(Y. Wu et al. 2018). Different structures of ANNs have been tested in this context, Deep Belief Networks (DBF)(Huang et al. 2014), Restricted Boltzmann Machines (RBM)(Tan et al. 2016), Recurrent Neural Network (RNN) (Yu et al. 2017) and more complex hybrid ANN models(Yi, Heejin, and Bae 2017). Long short-term memory (LSTM) network overperforms other algorithms in traffic prediction according to the research of (Ma et al. 2015). Most of these studies just focus on temporal features of traffic systems, spatial information is seldom considered. While correlations between spatial and temporal information of vehicles are salient inherent characteristic which can better interpret traffic evolutions.

To fill the gap, Ma et al. proposed a Convolutional Neural Network (CNN) framework to predict vehicular speed using taxi trajectory data in Beijing. This paper just aggregates speeds of floating vehicles to time-mean speeds of freeway segments, and feeds them as pixel variables into a 1-channel CNN, while other features such as flow and density are also important to reflect traffic states(Cassidy, Jang, and Daganzo 2012), simply omitting them will cause potential biases of regression models. To get a full-scale understanding of traffic, a 10.3-mile freeway segment in North California was chosen to be studied, partially observed traffic information (speed, flow, occupancy) were transformed into pictures, which are further fed into CNNs to predict travel time of that corridor in a time resolution of 5 minutes.

The rest parts of the paper are arranged as: 2 introduces the collection process of both traffic information data and travel time data of the target freeway segment. 3 describes how the data are preprocessed and transformed to images. Structures of CNNs are explained detailly in 4. The methodology and results are shown in 5. Conclusions and future work are listed in 6.

2 Data Collection

The experimental freeway segment is I-80 E from Davis(38.54122, 121.73476) to West Sacramento(38.57489, 121.57166), with a total length of 10.3 miles. This is a multi-lane corridor between San Francisco and Sacramento, two main cities in North California. Traffic demands are high there especially during afternoon peak hours, making it an appropriate scenario to look deep into the causes of bottlenecks, stop-and-go congestion and back propagation of shock waves(Zheng, Ahn, and Monsere 2010). Since considering every individual vehicle needs very large sample size of data and has great computing redundancy, aggregate data is analyzed in this paper to reflect macroscopic characteristics of the traffic flow. There are two main sources providing necessary data for the study: Caltrans Performance Measurement System (PeMS) and Bluetooth sensors.

2.1 Traffic information collection

California Department of Transportation (Caltrans) distributes detectors on freeways in California to monitor the operation of transportation systems and provide information for road users and researchers. These data has been made open source and can be extracted from http://pems.dot.ca.gov. The primary data source for the target freeway is the vehicle detector stations (VDS), providing real-time mean speed and flow of vehicles and occupancy data. Each VDS has loop detectors under individual lanes, sending back data to the server every 5 minutes. These data are aggregated to speed, flow and occupancy of road segments (one VDS corresponding to one segment) in a time resolution of 5 minutes, with a variable 'observation rate' indicating the health of VDSs. Observation rates range from 0 to 1, representing the confidence of data in each time interval, 1 means all the loop detectors in that VDS run in good condition, 0 means they all fail and 0-1 means traffic data is partially observed in that 5 minutes.

As shown in Figure 1, There are 15 VDSs along the I-80 E from Davis to Sacramento, totally 17 days data of them are downloaded from PeMS, 15 days (10/15/2018 – 10/29/2018) are treated as the training set, 2 days (10/31/2018 and 11/06/2018) will be validated by trained models, labelled as Type 1 and Type 2 testing sets correspondingly, Type 1 has the same observation rate matrices with the training set, while Type 2 is different. In another word, the detectors' health remained the same in October 2018, while changed in November. The reason why choosing two different types of testing sets is to test the generalization abilities of different CNN structures, which will be specified later.

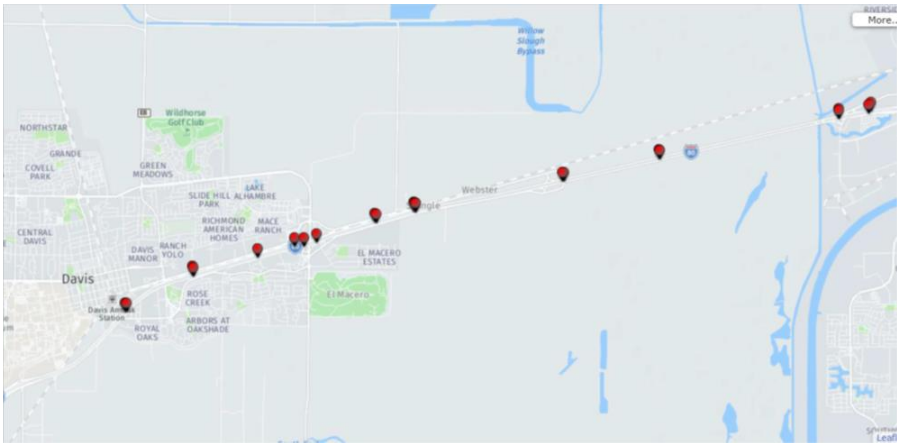


Figure 1 Positions of Vehicle Detector Stations

2.2 Travel time collection

Travel time data used in this study is provided by Bluetooth sensors of Iteris Inc. Bluetooth vehicle detectors are part of Vehicle-to-Infrastructure (V2I) system (Haghani et al. 2010), it facilitates the identification of vehicles containing Bluetooth devices, recoding their ids and passing time. Four Bluetooth sensors are installed, Sensor A at the starting point, D at the ending point of the 10.3-mile distance, B and C at the middle of the road, positions of which are listed in Figure 2.

Index	Corridor	Street Name	Latitude	Longitude
A	I-80	Richards Blvd O.C.	38.54122	121.7347596
B	I-80	Chiles/Rd 105D	38.55807	121.671181
C	I-80	Webster UC	38.56365	121.638873
D	I-80	Enterprise Blvd EO	38.57489	121.571657

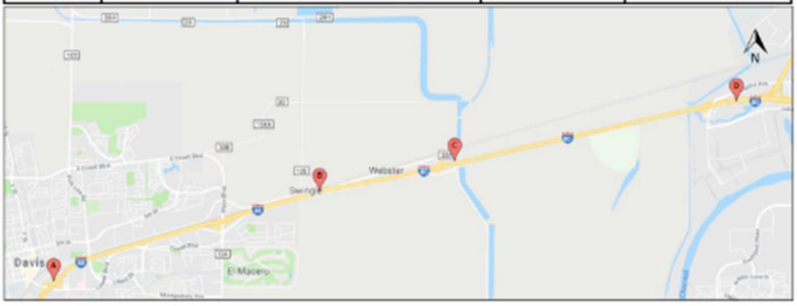


Figure 2 Positions of Bluetooth vehicle detectors

The travel time data stores raw records of Unix time stamps when vehicles passed 4 Bluetooth detectors paired with unique identifications, which call for further processing.

3 Data preprocessing and transformation

3.1 Travel time normalization

Differential time of passing the first and last Bluetooth detectors is calculated as individual vehicles' travel time. To omit outliers caused by telecommunication noise, only data in a standard deviation from the mean is considered, and the average value of them is set as the aggregate travel time of that 5 minutes. Figure 3 indicates that the majority of travel time distributes in the interval of [450s, 720s], to make the model converge faster and avoid potential regression bias, travel time in the interval of [<450], [450,480], [480,510], [510, 540], [540, 570], [600, 630], [630,660], [660, 690], [690, 720], [>720] (s) are categorized to 0-9 (10 categories). Therefore, the regression problem is transformed to a classification problem.

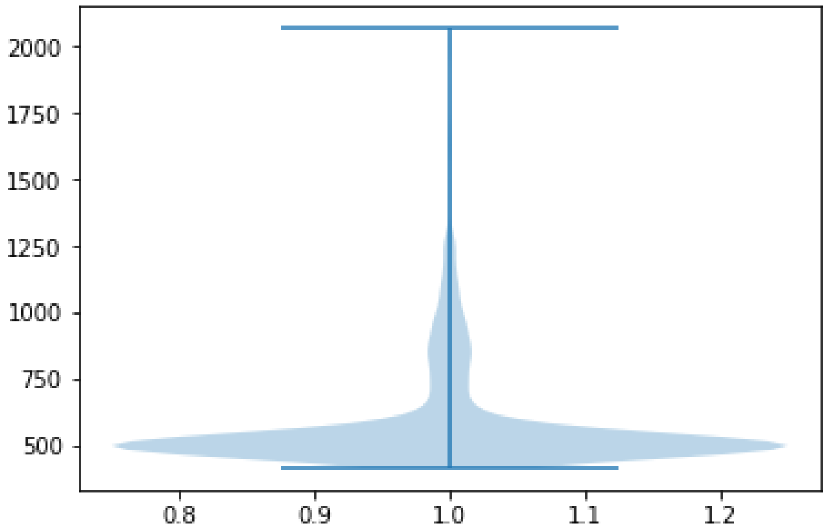


Figure 3 Distribution of travel time (s)

3.2 Time-space image generation

There are 4 variables used to predict travel time, namely, speed, flow, occupancy and observation rate of VDSs. To give an estimation of travel time in the next 5 minutes, the model takes 75 minutes of data before that. This 75-min data is arranged into a 15x15 matrix, each row corresponding to a VDS, each column corresponding to 5-min time interval. Then these matrices are converted to 15x15 images, with pixel values normalized to 0-1. Figure 4 is a demonstration of how these images look like, there are 4 sets of images in total, pixels of images in each set represents speed, flow, occupancy, observation rate correspondingly. Each image is paired with the travel time of the freeway segment in next 5 minutes, paired information is structed as a data frame for Pytorch training(Paszke et al. 2017).

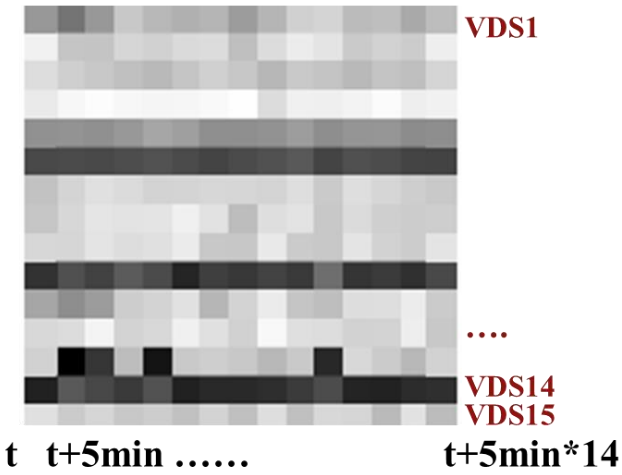


Figure 4 Sample images containing speed information of 15 VDSs for 75 minutes

4 Convolutional Neural Network

Convolutional Neural Network (CNN), is a widely used neural network architecture for pattern recognition and picture classification(Krizhevsky, Sutskever, and Hinton 2012). A typical CNN consists of convolutional layers, pooling layers and fully connected layers. The cooperation of these layers helps classify images more precisely with fewer neurons. There are 4 layers in my CNN(orderly): **I**.Convolutional layer 1 **II**.Max pooling layer **III**.Convolutional layer 2 **IV**.Fully connected layer, as shown in Figure 5.

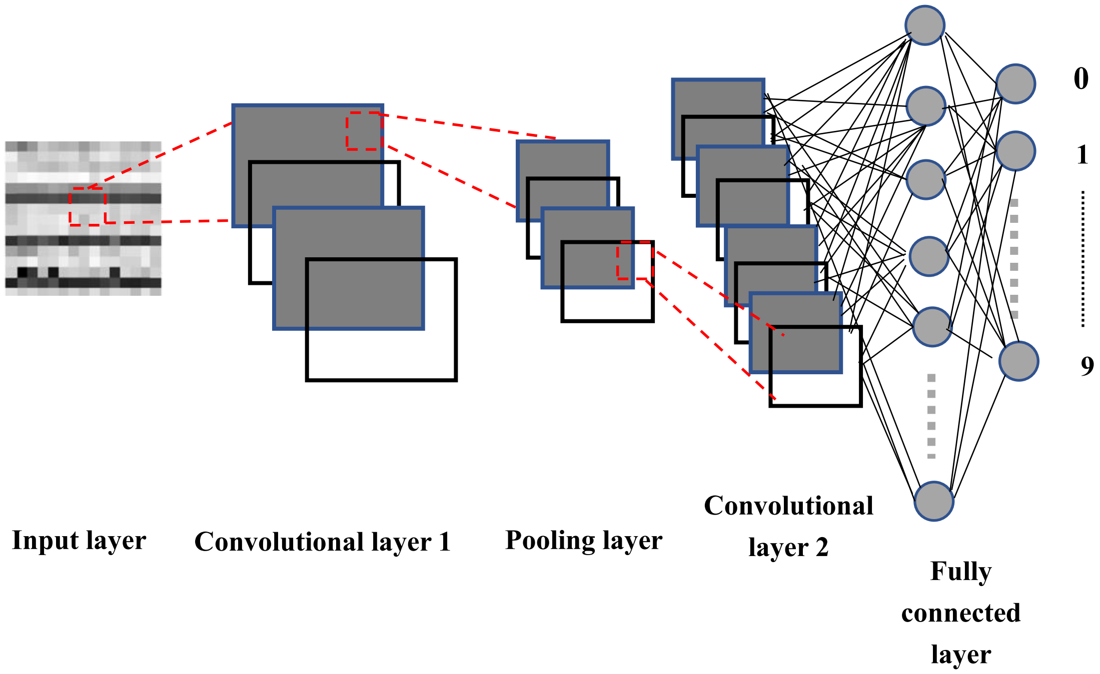


Figure 5 Structure of 1-Channel CNN

4.1 Convolutional layers

The input images are first processed by a convolutional layer, which uses multiple filters to detect specific patterns of a given image. The filters' size of the first convolutional layer in CNNs proposed in this paper is 3x3 with a stride of 1, and there are 3 types of filters, detecting bottlenecks, congestions and back propagation of shock waves correspondingly.

1	1	1
-1	-1	-1
-1	-1	-1

1	-1	-1
1	-1	-1
1	-1	-1

-1	-1	1
-1	1	-1
1	-1	-1

Figure 6 Three types of filters

The filters will be activated only when they detect desired features. After the convolution operation, the outputs of a 15x15 image are 3x13x13 feature maps, 3 for 3 filters.

Since the tensors processed by the first convolutional layer and the pooling layer will lose physical features, so there is no need to design specific filters for them, the filters used in the second convolutional layer is 20 stochastic ones.

4.2 Pooling layer

Pooling layers are designed to downsample and aggregate data because they only extract salient numbers from the specific region. The pooling layers guarantee that CNN is locally invariant, which means that the CNN can always extract the same feature from the input, regardless of feature shifts, rotations, or scales(LeCun, Bengio, and others 1995). The pooling mechanism used in this paper is Max Pooling(Masci et al. 2011), which takes the maximum value of a 2x2 grid. The 2x2 grid moves on the feature maps extracted by the convolutional layer with a stride of 2, so the feature size after Max Pooling is 3x7x7, which needs fewer parameters compared with fully connected ANNs.

4.3 Fully connected layer

Layers in the CNN are connected by rectified linear units (ReLU)(Dahl, Sainath, and Hinton 2013), which ensures nonlinearity of the model and makes the backpropagation process of neural networks feasible(Figure 7).

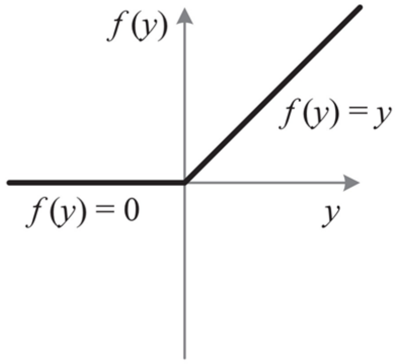


Figure 7 Rectified Linear Unit

The features obtained by convolutional layers with max pooling mechanism will be (20x5x5) tensors, then these tensors are flattened into 1000x1 vectors, fed into a fully connected layer. The output size of the fully connected layer is 10, corresponding to (0-9) categories. The drop out mechanism is introduced here(Sutskever et al. 2014): each neuron in the fully connected layer has a probability(set as 0.5) to be dropped out during each batch of training, to eliminate a potential bias of the model. Stochastic Gradient Descent (SGD) for weights update with cross entropy loss function is applied,

$$loss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

M is the number of classes (10 here), y is a binary indicator (0 or 1) if class label c is the correct classification for observation o, p denotes predicted probability observation o is of class c.

5 Model training

5.1 1-Channel CNN

1-Channel CNN means only images representing single variables (speed/flow/occupancy) are fed into the neural network, and the pixel values of these images are partially observed since not all the VDS run in good condition, which can be reflected by their observation rates.

First step is to tune the parameters of CNN. The basic architecture has been described in 4, using stochastic gradient descent to update the weights, there are 2 parameters need tuned: the batch size and the learning rate. Figure 8 is the training loss over epochs with different model parameters of 1-Channel CNN (speed), which suggests, with a batch size of 10 and a learning rate of 0.01, the model converges fastest, thus this combination of parameters is set.

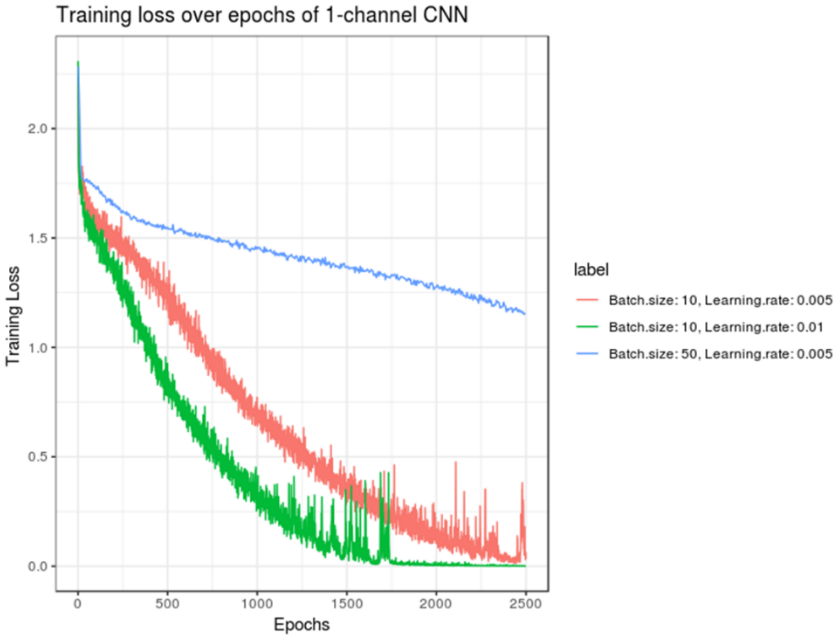
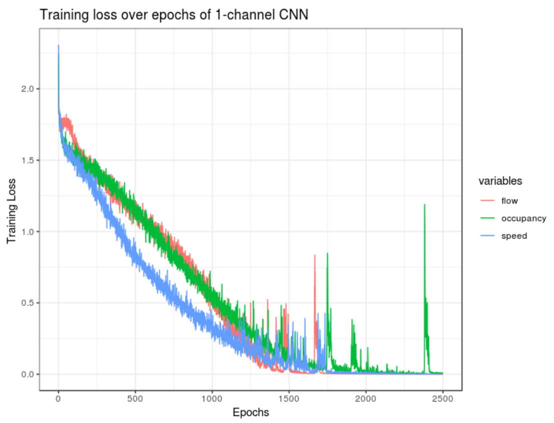


Figure 8 Training loss of different batch sizes and learning rates



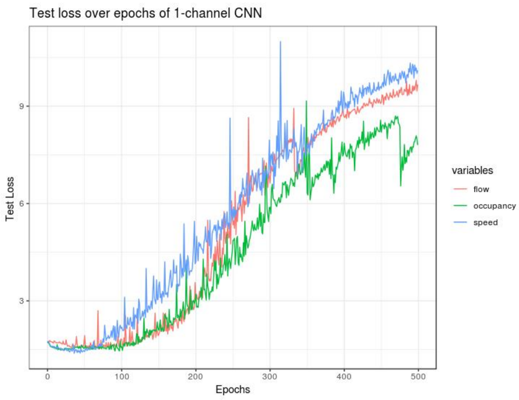


Figure 9a Training loss of 1-Channel CNN **Figure 9b Test loss of 1-Channel CNN**

Figure 9 shows the 1-Channel CNN has a good fitting capability on the training set that MSE training loss converges to 0, while the curve of test loss on the Type 1 testing set indicates the model tends to overfit since test loss goes up as epochs increase. Therefore, a real-time accuracy algorithm is applied: it records the prediction accuracy (0-100%) on the testing set of each training epochs, to tell when the model performs best without overfitting. Figure 10 demonstrates the prediction accuracy of 1-Channel CNNs, we can summary that the three different variables have similar highest predition accuracy, among them occupancy has the best performance: reaches the highest accuracy first and has the lowest overfitting tendency.

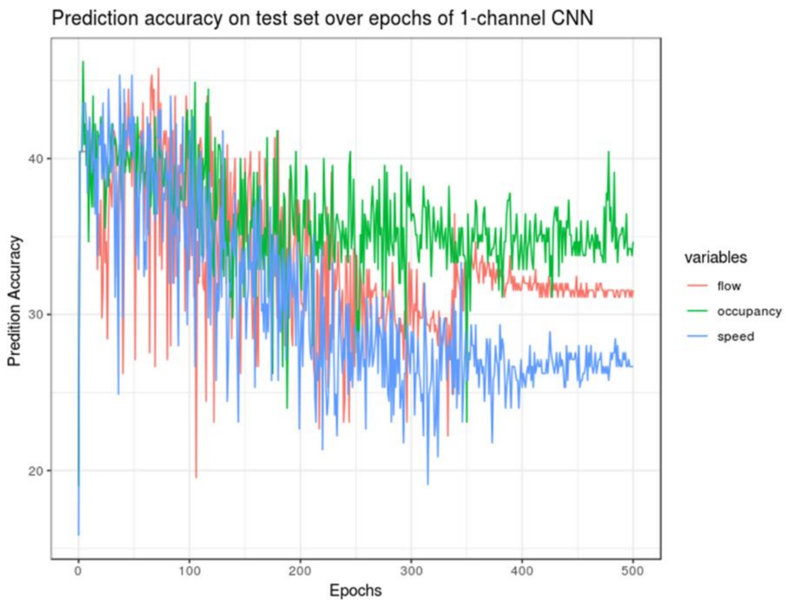


Figure 10 Prediction accuracy on Type 1 testing set of 1-Channel CNN

5.2 3-Channel CNN

To take a full usage of data, speed, flow, occupancy, the three fundamental features in traffic flow theory are fed into the CNN simutaneously. During each batch of training, there are 10 samples flowing into the model, each of them has 3 images, representing speed, flow and occupancy correspondingly. Figure 11 shows the 3-Channel model overperforms the 1-Channel model with a 39.6 % (46.0% to 64.2%) increase of prediction accuracy. While Figure 12 indicates the 3-channel model does not perform as good on the Type 2 testing sets(11/06) compared with Type 1(10/31). It suggests the model has a poor generazation ability on the testing sets having different observation rates with the training sets.

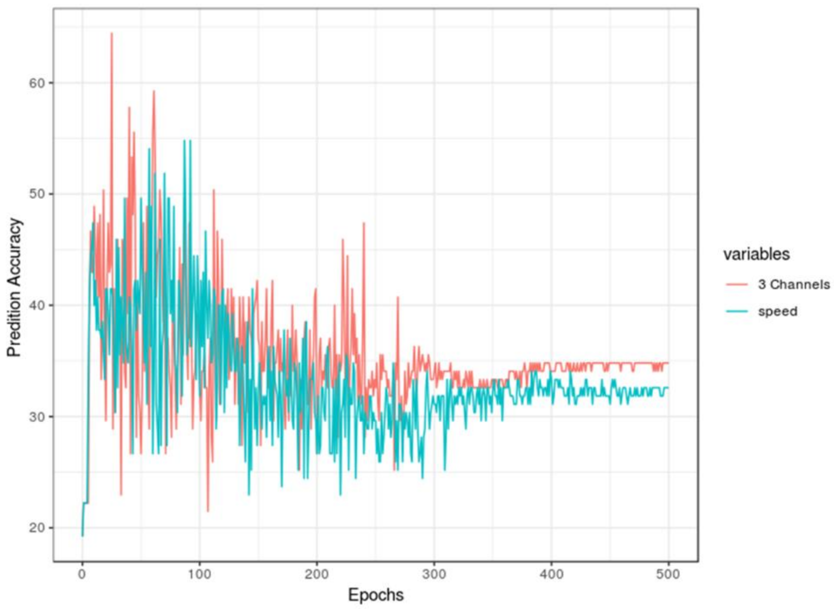


Figure 11 Prediction accuracy on Type 1 testing set of 1-Channel and 3-Channel CNN

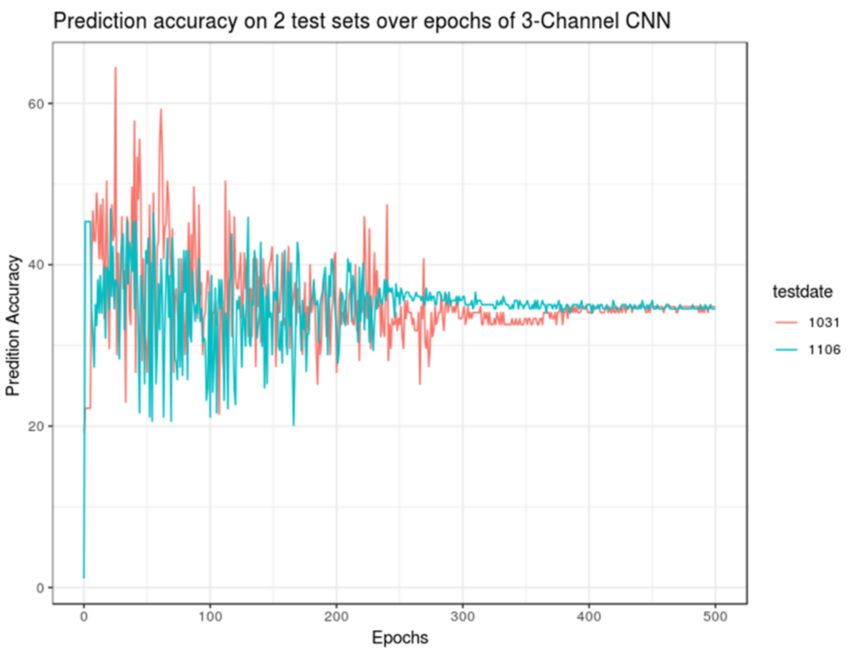


Figure 12 Prediction accuracy on Type 1 and Type 2 testing sets of 3-Channel CNN

5.3 4-Channel CNN

The unsatisfactory generazation of 3-Chaneel CNN calls for a more precise model, thus the observation rates of detectors are fed into the CNNs as an additional channel. The 4-Channel CNN is able to handle partially observed data since it takes the observation rates into consideration, which reflect how good is the data. Figure 13 shows the 4-Channel CNN has a higher prediction accuracy on the Type 2(11/06) testing set compared with 3-Channel models with an increas of 20.6% (47.1% to 56.8%).

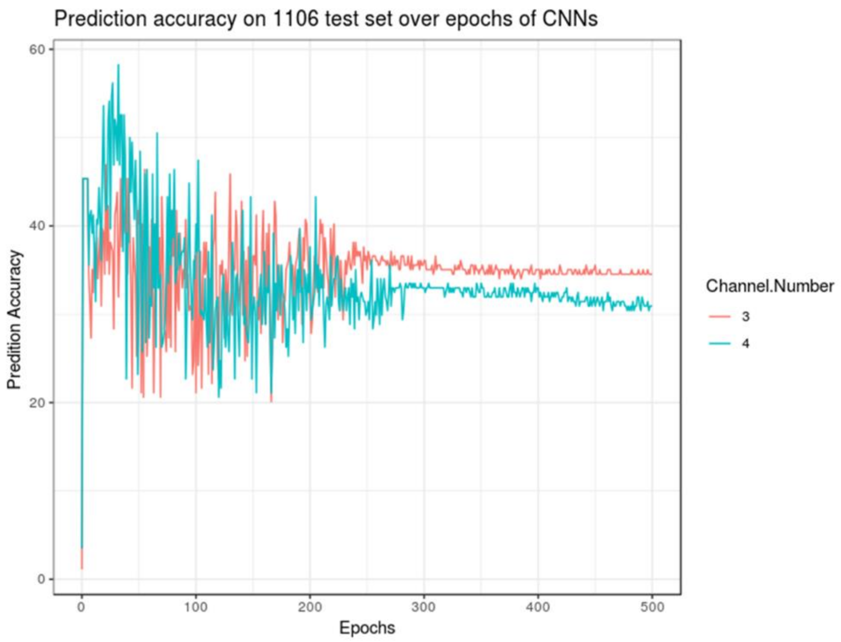


Figure 13 Prediction accuracy on Type 2 testing sets of 3-Channel CNN and 4-Channel CNN

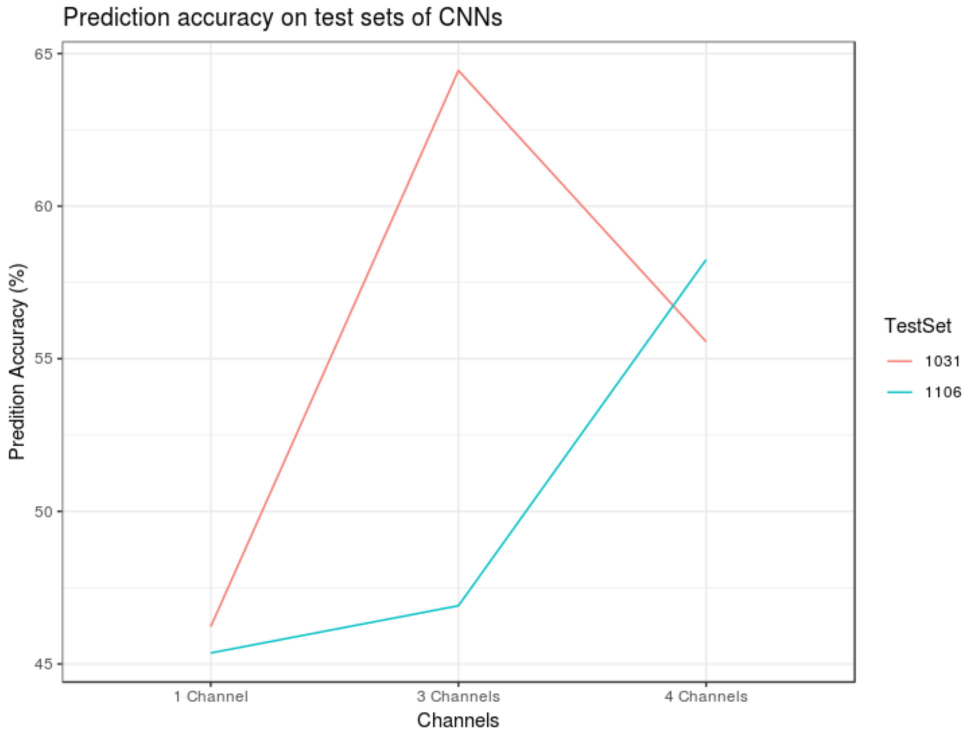


Figure 14 Prediction accuracy on Type 1and Type 2 testing sets of CNNs

6 Conclusion and Discussion

Transforming traffic information to images makes the model handle temporal and spatial data simultaneously, which gets a thorough understanding of traffic evolution. The results in Figure 14 shows 3-Channel CNN has the highest prediction accuracy on Type 1 testing set, while 4-Channel CNN performs best on Type 2 testing sets. With the number of input channels increases, the CNN model has better generazation on testing sets having different data quality with the training set, which means the model has a better ability to process partially observed data.

There are 2 potential problems of the proposed models, which call for future study. **First**, the resolution of traffic data is 5 minutes due to the restriction of data source, making 15x5 minutes a too large time interval for travel time prediction, since traffic can change a lot in 75 minutes. Thus analysis on time-continuous data will be a better choice, like trajectories of Uber/Lyft cars, the Max Pooling mechanism in CNNs helps downsamples images, which is promised to handle images with large scales. **Second**, The cross entropy loss function is not so reasonable in the context of travel time prediction as the distance from the right class makes sense here. Missclassifying 2 to 1 and 5 should be assigned different weights of punishment, other than uniform 0-1 loss.

Code:

https://github.com/JeanUCD/CNN_traveltime_prediction

References

Burghout, Wilco, Haris Koutsopoulos, and Ingmar Andréasson. 2007. "Hybrid Mesoscopic-Microscopic Traffic Simulation." *Transportation Research Record: Journal of the Transportation Research Board*.

Cassidy, Michael J., Kitae Jang, and CF Daganzo. 2012. "Macroscopic Fundamental Diagrams for Freeway Networks." *Transportation Research Record*.

Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton. 2013. "Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout." In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*,.

Davis, Gary A., and Nancy L. Nihan. 2007. "Nonparametric Regression and Short-Term Freeway Traffic Forecasting." *Journal of Transportation Engineering*.

Fei, Xiang, Chung Cheng Lu, and Ke Liu. 2011. "A Bayesian Dynamic Linear Model Approach for Real-Time Short-Term Freeway Travel Time Prediction." *Transportation Research Part C: Emerging Technologies*.

Haghani, Ali et al. 2010. "Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors." *Transportation Research Record: Journal of the Transportation Research Board*.

Helbing, D., A. Hennecke, V. Shvetsov, and M. Treiber. 2002. "Micro- and Macro-Simulation of Freeway Traffic." *Mathematical and Computer Modelling*.

Hofleitner, Aude, Ryan Herring, and Alexandre Bayen. 2012. "Arterial Travel Time Forecast with Streaming Data: A Hybrid Approach of Flow Modeling and Machine Learning." *Transportation Research Part B: Methodological*.

Huang, Wenhao, Guojie Song, Haikun Hong, and Kunqing Xie. 2014. "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning." *IEEE Transactions on Intelligent Transportation Systems*.

Jiann-Shiou Yang. 2005. "Travel Time Prediction Using the GPS Test Vehicle and Kalman Filtering Techniques."

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *ImageNet Classification with Deep Convolutional Neural Networks*,.

Kwon, Jaimyoung, Benjamin Coifman, and Peter Bickel. 2007. "Day-to-Day Travel-Time Trends and Travel-Time Prediction from Loop-Detector Data." *Transportation Research Record: Journal of the Transportation Research Board*.

LeCun, Yann, Yoshua Bengio, and others. 1995. "Convolutional Networks for Images, Speech, and Time Series." *The handbook of brain theory and neural networks*.

Lindveld, Charles D. R., Remmelt Thijs, Piet H. L. Bovy, and Nanne J. Van der Zijpp. 2007. "Evaluation of Online Travel Time Estimators and Predictors." *Transportation Research Record: Journal of the Transportation Research Board*.

van Lint, J.W.C. 2004. Proceedings of the 17th international conference on Hybrid

systems: computation and control - HSCC ’14 “Reliable Travel Time Prediction for Freeways.”
Ma, Xiaolei et al. 2015. “Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data.” *Transportation Research Part C: Emerging Technologies*.
Masci, Jonathan, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction.” In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*,.
Mori, Usue, Alexander Mendiburu, Maite Álvarez, and Jose A. Lozano. 2015. “A Review of Travel Time Estimation and Forecasting for Advanced Traveller Information Systems.” *Transportmetrica A: Transport Science* 11(2): 119–57. https://doi.org/10.1080/23249935.2014.932469.
Oh, Cheol, Stephen Ritchie, and Jun-Seok Oh. 2007. “Exploring the Relationship Between Data Aggregation and Predictability to Provide Better Predictive Traffic Information.” *Transportation Research Record: Journal of the Transportation Research Board*.
Paszke, Adam, Sam Gross, Soumith Chintala, and Gregory Chanan. 2017. “PyTorch.” *GitHub Repos.*
Sutskever, Ilya, Geoffrey Hinton, Alex Krizhevsky, and Ruslan R Salakhutdinov. 2014. “Dropout : A Simple Way to Prevent Neural Networks from Overfitting.” *Journal of Machine Learning Research*.
Tan, Huachun et al. 2016. “A Comparison of Traffic Flow Prediction Methods Based on DBN.”
Tenenboim, Einat, and Yoram Shiftan. 2018. “Accuracy and Bias of Subjective Travel Time Estimates.” *Transportation*.
Wu, Chun Hsin et al. 2003. “Travel Time Prediction with Support Vector Regression.” In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*,.
Wu, Yuankai et al. 2018. “A Hybrid Deep Learning Based Traffic Flow Prediction Method and Its Understanding.” *Transportation Research Part C: Emerging Technologies*.
Yi, Hongsuk, Jung Heejin, and Sanghoon Bae. 2017. “Deep Neural Networks for Traffic Flow Prediction.” In *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*,.
Yu, Haiyang et al. 2017. “Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks.” *Sensors (Switzerland)*.
Zhang, H. M. 2002. “A Non-Equilibrium Traffic Model Devoid of Gas-like Behavior.” *Transportation Research Part B: Methodological*.
Zhang, Xiaoyan, and John A. Rice. 2003. “Short-Term Travel Time Prediction.” *Transportation Research Part C: Emerging Technologies*.
Zheng, Zuduo, Soyoung Ahn, and Christopher M. Monsere. 2010. “Impact of Traffic Oscillations on Freeway Crash Occurrences.” *Accident Analysis and Prevention*.