Highway Traffic Characteristics Estimation using Deep Convolutional Neural Network

2019-04-26

Zili Ge
Division of Engineering Science, University of Toronto,
iCity Centre for Automated and Transformative Transportation Systems (iCity-CATTS)
Email: zili.ge@mail.utoronto.ca

Ahmed Aqra
Department of Civil Engineering, University of Toronto,
iCity Centre for Automated and Transformative Transportation Systems (iCity-CATTS)
Email: ahmed.aqra@mail.utoronto.ca

Paper prepared for presentation
at the **Artificial Intelligence and Machine Learning for Smart Mobility** Session
Of the 2019 TAC-ITS Canada Joint Conference, Halifax, NS

This work is made possible through the National Sciences and Resarch Council of Canada.
Application Number: USRA – 526527
We gratefully ackowledge support provided by the Ontario Ministry of Research and Innovation through the ORF-RE program.

1 ABSTRACT

Recent watershed advancements in computer vision and machine learning has allowed for the possibility of classifying traffic characteristics from camera image data in real time. Information from traffic images can supplement data from other sensors such as loop detectors, Bluetooth and WiFi sensors and Dedicated Short Range communications (DSRC) roadside units. In this paper, we propose a method for near real-time estimation of traffic state variables such as volume and speed, in locations where traffic cameras exist. The proposed system allows municipalities and provinces to extend the utility of their existing camera systems with minimal additional resources. In this paper, we specifically explore the application of convolutional neural networks (CNN) to traffic image processing. We use existing loop detector data from Toronto highways as the ground truth to train and test the CNN to infer the macroscopic traffic flow characteristics of speed and flow from the still images. Preprocessing using temporal median stacking and image subtraction was first done to identify cars in lanes. The model was then trained, using ground truth data from loop detectors, to estimate traffic speed and volume directly from the images for all vehicle types. The proposed model generates promising results, with up to 88.6 percent accuracy, depending on the bin size.

Keywords: CNN, Deep Learning, Traffic Detection and Estimation, Density, Traffic Monitoring, Traffic Cameras, Big Data, Articial Intelligence, Data-Driven Model, Traffic Flow.

2 INTRODUCTION

The proliferation of big data has seen its applications to traffic monitoring and control. Research has demonstrated the possibility of real-time traffic detection and estimation using both image and video analytrics(1)(2)(3). Existing methods for image classification, however, are graphics intensive and computationally expensive which may make it difficult for many municipalities to apply to widely existing traffic cameras. There is work that attempts to integrate heterogeneous data sources, although they do not explicitly cross validate each other (3). Other than video data, there has also been work on the extraction of low-level features and traffic parameters, such as density, for both traffic and crowd control (4)(5)(6)(8).

For traffic counting, loop detector data is generally considered more accurate than camera data (3). There is work in the area of comparing loop detector data to camera data; however, there is little precedent for correlating loop detector data with traffic camera data. An intensive literature analysis has not found any prior work in this area.

The model presented in this paper provides added value to existing traffic monitoring tools by providing a tool to detect various traffic characteristics that were previously restricted to loop detectors, which are often unreliable; for instance, in this study, out of the 74 loop detectors that were used, only 48 were found to be producing acceptable data. In addition, loop detectors are costly to repair and its maintenance requires civil work and road closures, as replacing loop detectors involves replacing asphalt. Cameras are widely used for traffic monitoring, detection and even for security; as a result, there are billions of unlabelled images that a model can gather traffic data from. The proposed model has the ability to label these images automatically for future models that require more data points.

In addition, this work will be used to classify traffic information on a macroscopic fundamental diagram (MFD). Traffic is either in the state of free flow, bound or congested, as shown in Figure 1. The flow q is defined as:

$$q=ku_0-\frac{k^2u_0}{k_j} \tag{1}$$

where u_0 is the speed of traffic at free-flow, k is the density, defined as vehicles per length and k_j is the jam density, or the density in which the traffic is at a standstill. This follows a parabolic relationship that this model attempts to capture.

We attempt to capture traffic flow characteristics, however, from a data-driven perspective. We hope to replicate some of the characteristics present in a mathematical approach, such as the parabolic relationship displayed in the MFD, through implicit relationships learned by the model.

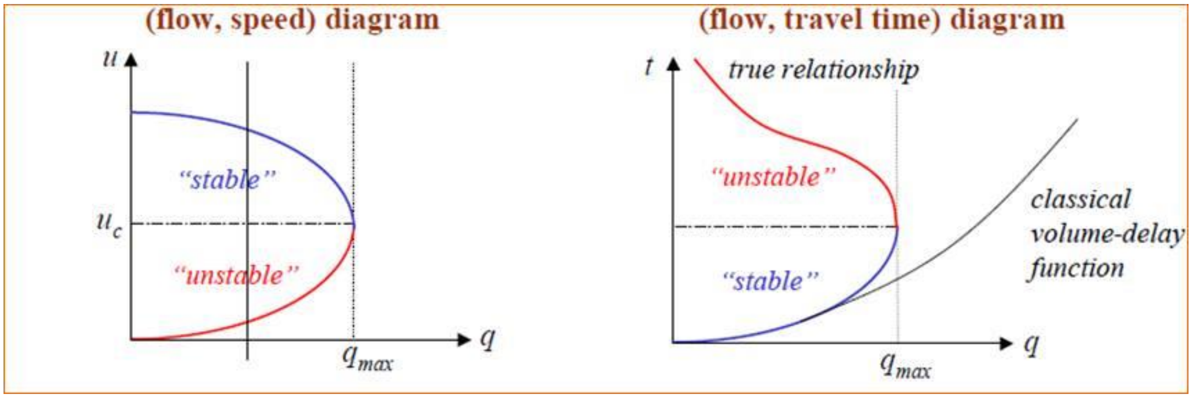


FIGURE 1 Macroscopic Fundamental Diagram (10)

3 METHODOLOGY

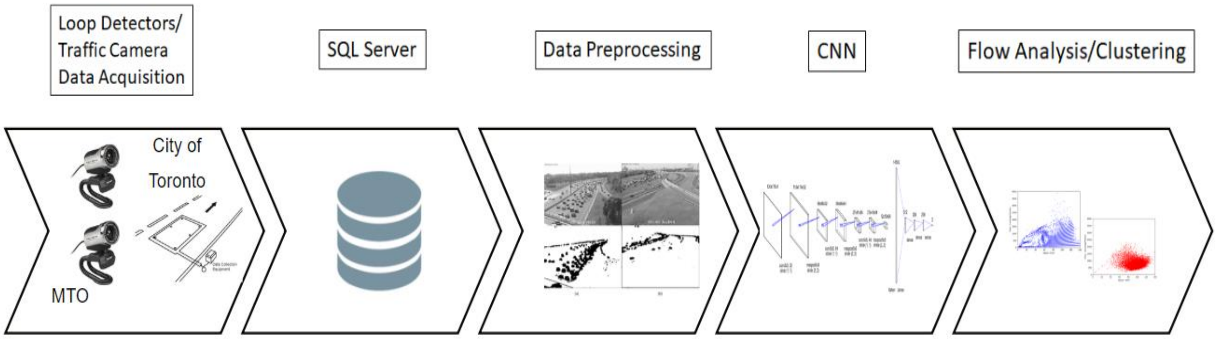


FIGURE 2 Model Architecture and Data Sources

We attempt to correlate data at different spatiotemporal scales; loop detectors provide microscopic data for individual vehicles, while camera images are captured every 3 minutes and do not possess low-level information. This paper attempts to reconcile the two regimes by processing loop detector data to determine the macroscopic properties of speed and flow.

In order to match the data sources temporally, loop detector data was averaged in order to match the frequency of collection of camera data. The assumption was made that nearby loop detectors correspond to nearby cameras, as both the loop detectors and cameras operate on the highway. There is some difficulty in correlating both spatially, due to the lack of information regarding loop detector and camera directionality; this is an area of further work and is explored further in the conclusion. The main focus of this paper is to demonstrate the process of correlating data from disparate sources to make them consistent in the creation of a multimodal sensor network, where data is acquired from multiple sensors in order to construct a more robust model of traffic congestion.

3.1 Data Acquisition

We acquired our training data through *On511*, an open data initiative that provides near-real time updates from traffic cameras across Ontario, Canada. For the sake of this study, only cameras in the Greater Toronto Area are considered, giving a total of 135 usable cameras. A Python script is used acquire the images and store them to a centralized relational database server. The images are updated every 3 minutes. Each image consists of a 704 × 480 × 3 tensor; the images have a resolution of 704 × 480, with 3 color channels.

We attempt to match camera information with data from loop detectors obtained from the City of Toronto and the MTO to provide our validation set. Euclidean distances between traffic cameras and loop detectors are calculated, with the closest loop detector to a camera considered the corresponding loop detector. Only cameras with a loop detector closer than 250 metres to the camera is considered. Relevant features are extracted in the preprocessing step, including vehicle detection.

3.1.1 Loop Detectors

Raw loop detector data is collected from the City of Toronto and the Ministry of Transportation from 1323 loop detectors for the May of 2018. A processing script is used to take the raw data and process it into speed and occupancy information, updated every 20 seconds. Information from 1323 detectors was provided by the city.Out of the 1323 detectors, 361 detectors did not have a valid GPS coordinate attached after postprocessing. These detectors were discarded, as it would be difficult to correlate the detector with a camera without prior knowledge on detector location.

To find the loop detectors that correspond to camera locations, we use Euclidean distances in order to correlate the two. A radius of 250m was used for this classification. Due to lack of information in the API call from *On511*, we were unable to correlate the direction of where the camera is facing with the loop detectors. The cameras are mounted on a swivel, which means that controllers could move the cameras at any time and the information on camera direction is not encoded. A radius of 250m was chosen; the number of cameras with a corresponding loop detector within a certain radius can be found in Table 2.

Visual inspection has found some of the loop detectors to be defective; they were returning null or nonsensical values for the duration of their operation. Due to the faulty sensors, a checksum of traffic velocity for the first day as well as the first reading of each day in a given month is first calculated. Any irregularities is disregarded, and the data is not used. It was found that 48 cameras had a corresponding loop detector. Out of the 48 cameras, 1 had an aspect ratio slightly different than the rest and was therefore resized and cropped. Another had an aspect ratio significantly different than the rest and was therefore discarded, leaving 47 cameras. Data imputation was considered, but was unsuitable for this application, as a sensor would have no viable data to impute (11) .

To convert microscopic data from loop detectors to speed and occupancy, we define occupancy as

$$\text{occupancy} = 100 \cdot \frac{\sum t_i}{\mathrm{T}} \qquad (2)$$

where t_i is the time there is a car over a detector, and T is the detection interval, which is windowed at 20s. The definition of speed is given as

$$speed = \frac{1}{N}\sum_{T} v_i \qquad (3)$$

where T is a window of 20 seconds, and N is the number of vehicles to pass over the detector at that given time.

A window of speed and occupancy readings from the loop detector is taken from 5 minutes prior to 1 minute after each image in order to account for missing datapoints, using

$$\frac{1}{n_{r,viable}} \sum_{\mathrm{i}\,=\,-5,\ r_i \neq 0}^{1} r_i \qquad (4)$$

to average out the reading and give a better overview. A nonzero reading is considered viable.

Since there is no simple relationship between occupancy and density, due to heterogenous traffic flow and varying vehicle sizes (12) . However, given the highway regime, a homogenous flow is assumed, and vehicle sizes are assumed constant. Under this assumption, the equation simplifies to

$$occupancy=(l+d)k \qquad (5)$$

where k is the flow, l is the car length and d is the width of the loop detector. Note that this differs from density by a constant; we then multiply this value by the speed of traffic *u* in order to approximate the traffic flow, given in units of % *coverage·hour/km*.

3.1.2 *Camera Traffic Images*

Data was collected from May 25, 2018 to May 31, 2018. For camera data, we only consider images from 6:00AM to 8:00PM, as the quality of the cameras means that images are overexposed and washed-out in low light conditions outside this time range. A total of 43,499 images were collected this way. Speed and occupancy data from the loop detectors are gathered every minute, for a total of 130,497 records. We also use a binary threshold to disregard any images that had issues in pre-processing, which leaves us with 39,138 images.

For data augmentation, the images are flipped horizontally as the traffic condition is expected to be invariant to road directionality. The preprocessing step sometimes has issues with changes in light intensity or changes in perspective due to camera swivel; this is filtered by counting the total number of pixels and removing all images that exceed a certain threshold. A total of 78,276 images are used for the training set after data augmentation.

Table 1 : Number of images during different stage of data processing

Stage of Processing	# of Images
Images in database	436629
Images with loop detector	43499
Images after thresholding	39138
Images after augmentation	78276

We matched the camera information with data from loop detectors obtained from the City of Toronto and the MTO to provide our validation set. Euclidean distances between traffic cameras and loop detectors are calculated, with the closest loop detector to a camera considered the corresponding loop detector. Only cameras with a loop detector closer than 250 metres to the camera is considered. Relevant features are extracted in the preprocessing step, including vehicle detection.

Table 2 : Number of cameras with corresponding loop detector as a function of radius

Threshold Radius (m)	# of Cameras
100	33
150	61
200	71
250	74
300	76
500	79

3.2 Preprocessing

Procedure 1 Median Stacking Algorithm.

```
procedure Median($f_i$)
    Load images $f_{i-12}$ to $f_{i+12}$ in greyscale
    $f_{median} \leftarrow$ median value for all pixels from $f_{i-12}$ to $f_{i+12}$
    for images from $f_{i-2}$ to $f_{i+2}$ do
        $f_i \leftarrow |p_{median} - p_i|$ for all pixels in image
        Binary threshold, smooth $f_i$
        Downsample $f_i$
    return $f_i$
```

We first apply temporal median stacking and image subtraction in order to find cars as demonstrated in Procedure 1**Error! Reference source not found.**. A window is obtained to construct an image where every pixel is a median value as the ground, which is subtracted from the image of interest, thresholded and blurred in order to capture nonstationary objects from frame to frame. As this is a highway setting, all nonstationary objects are assumed to be vehicles. Various characteristics of the traffic cameras are exploited to develop a quick, accurate and scalable algorithm for car detection on stationary, low resolution highway cameras. This step is robust for faraway vehicles, as the model does not involve explicit vehicle classification. A similar motivation was explored in (7), with the exception that preprocessing was done through CNNs rather than temporal based median stacking, as the assumption of a fixed camera reference point cannot be made.

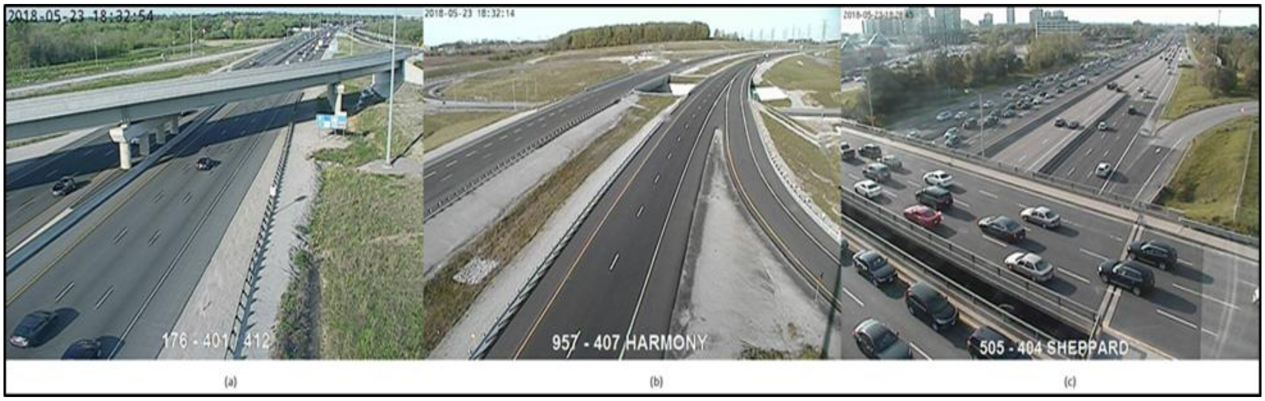


FIGURE 3 Examples of problem cases. (a) shows a bridge over the highway, while (b) shows a prominent ramp, as well as a street running below the highway. (c) is an example where a main arterial blocks off significant amounts of the foreground.

The motivation is merely to remove certain features from the training model; this image segmentation technique is to extract relevant features for the input to the model and not for car detection (8).

4. Model Framework

We explore the use of a Convolutional Neural Network (CNN) in the classification of images. The preprocessed image serves as the input layer of Figure **Error! Reference source not found.**. ReLU activation is used for the activation functions; this is fairly standard for a CNN.

Mean square error is used as the loss function, defined as

$$\frac{1}{n}\sum_{i}(v_{i,pred,occ}-v_{i,label,occ})^{2}+(v_{i,pred,speed}-v_{i,label,speed})^{2} \tag{6}$$

The two output neurons output floats for speed and occupancy, the two relevant features extracted from the training set. The activation for the final layer is a linear activation function, as floats are predicted. Softmax is not appropriate as this is not a classification problem. Speed and density are normalized so the output neurons output values in a similar range, so the mean squared error does not ignore one characteristic over another. To calcuate the accuracy, the function below is

$$\frac{1}{n}\sum_{i}\frac{1}{2}\frac{|d_{i,pred}-d_{i,loop}|}{|d_{i,loop}|+|d_{i,pred}|} \qquad (7)$$

with independent accuracies for the 2 different predicted values are used. We do not use standard error, as the ground truth is occasionally zero; this would give an undefined standard error. As a result, we sum the predicted and labelled values, then divide by two.

During the design process four different models were developed: Regular, shallow, deep and Regular with fiters. Table 3 below shows the defintion of different haypermeteres and configurations used for the models design.

Table 3 Hyperparameter defintions

Haypermeter	Description
Layers	The number and description of each layer. Convolutional layers followed by fully connected layers.
Filter Sizes	The size of the filter in which the algorithm is applied to in each convolutional layer
Stride	How much the filter moves each time
Neurons (Fully Connected)	The amount of neurons in each fully connected layer
Filter Amount	How many filters in each convolution layer

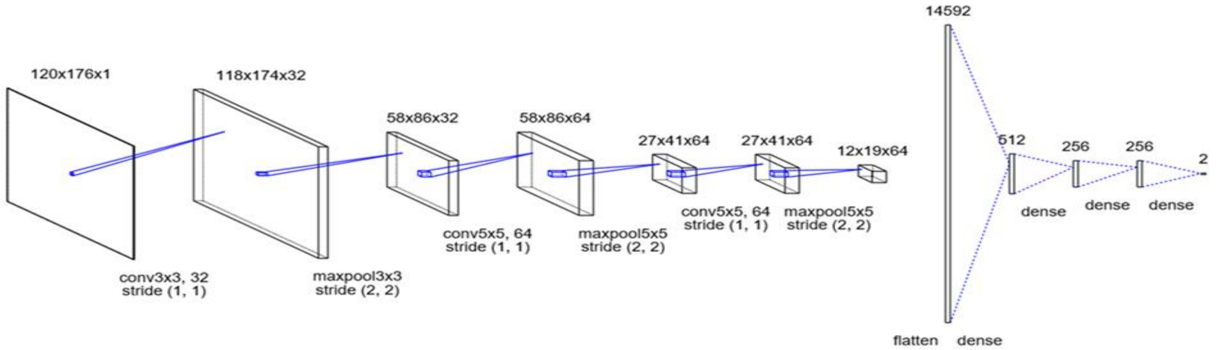


Figure 4 : Hyperparameter values for the Regular model design.

In the original model, three convolutional layers are used, followed by 3 fully connected layers. The convolutional layers has 32, 64, and 64 filters, respectively, with filter sizes of 5x5, 5x5 and 3x3. The images are zero-padded, with 2x2 max-pool after each convolutional layer. The output is then flattened and connected to three fully connected layers with 512, 256 and 256 neurons, respectively.

Several CNN variations were attempted in the classification of this task. In addition to the model currently in use, a shallower model with 2 fully connected layers and 2 convolutional layers were used. Different filter sizes were used as well; a third model uses filter sizes of 7x7, 5x5 and 3x3 for the three convolutional layers, with the motivation that larger features should be recognized earlier in training. A 4th model uses the filter size modification, in addition to using 5 fully connected layers as opposed to 3, each additional layer with 256 neurons each. Table 4, shows the charatristics of the four developed models and its hyperparameter values

Table 4 Hyperparameter values for different model configurations

Item	Layers	Filter Sizes	Stride	Neurons (Fully Connected)	Filter Amount
Regular	3 convolutional, 3 fully connected	5x5, 5x5, 3x3	2x2	512, 256, 256	32, 64, 64
Shallow	2 convolutional, 2 fully connected	5x5, 3x3	2x2	512, 256	32, 64
Deep	3 convolutional, 5 fully connected	7x7, 5x5, 3x3	2x2	512, 256, 256, 256	32, 64, 64
Filter	3 convolutional, 3 fully connected	7x7, 5x5, 3x3	2x2	512, 256, 256	32, 64, 64

Since the decision was made to use a CNN, as the classification is image-based, the constraints on training resources means that a larger model would take significant amounts of time to train.In addition, image preprocessing yielded a distinctive texture as can be seen o Figure **Error! Reference source not found.** (a) for at-capacity or (d) for free-flow; it is hoped that the CNN will be able to implicitly perform blob classification based on the image as shown in Figure 6.

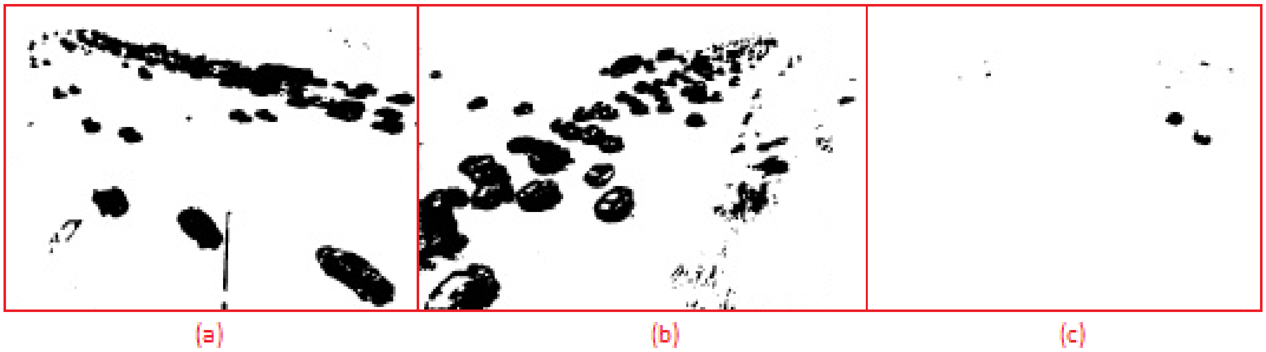


FIGURE 5 Car detection at different regimes. (a): Congestion (Speed = 26km/hr, Occupancy = 39%) – (b): Bound flow (Speed = 87km/hr, Occupancy = 10%) (c): Free flow (Speed = 97km/hr, Occupancy = 3%

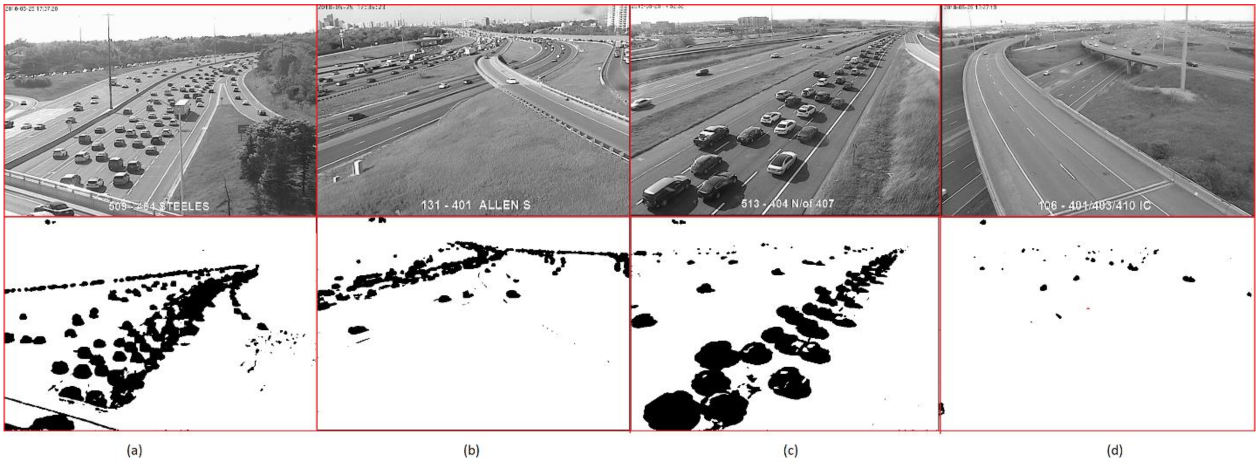


FIGURE 6 Before and after images for several traffic situtation.

5 RESULTS

5.1 The Computing Environment

The model is trained on a Nvidia GTX 950M, with a train time of approximately 5 minutes. 65,000 images were in the training set while 13,248 were used in the validation set. The size of the validation set stems from the rate of convergence of the loss function; the loss appeared to decrease throughout the training process; future work could be done to train the network on a larger corpus of data. The amount of neurons were decided largely in part by the amount of available memory in the network; training took place using 8GB of memory, and storing a larger model would require access to a more powerful machine.

5.2 Model Performance

5.2.1 Loss Function

The training demonstrated a loss function that followed a downward trend. However, due to the definitions of the loss and accuracy functions, predictions as given in Figure 7 and Figure 10 are dependent on the closeness of the float values. This means that the loss and accuracy functions can fluctuate significantly based on one wrong prediction in a batch. This can be seen in the high variability of accuracy in each batch. Figure 7 below shows the lost function for the four model design alternatives.

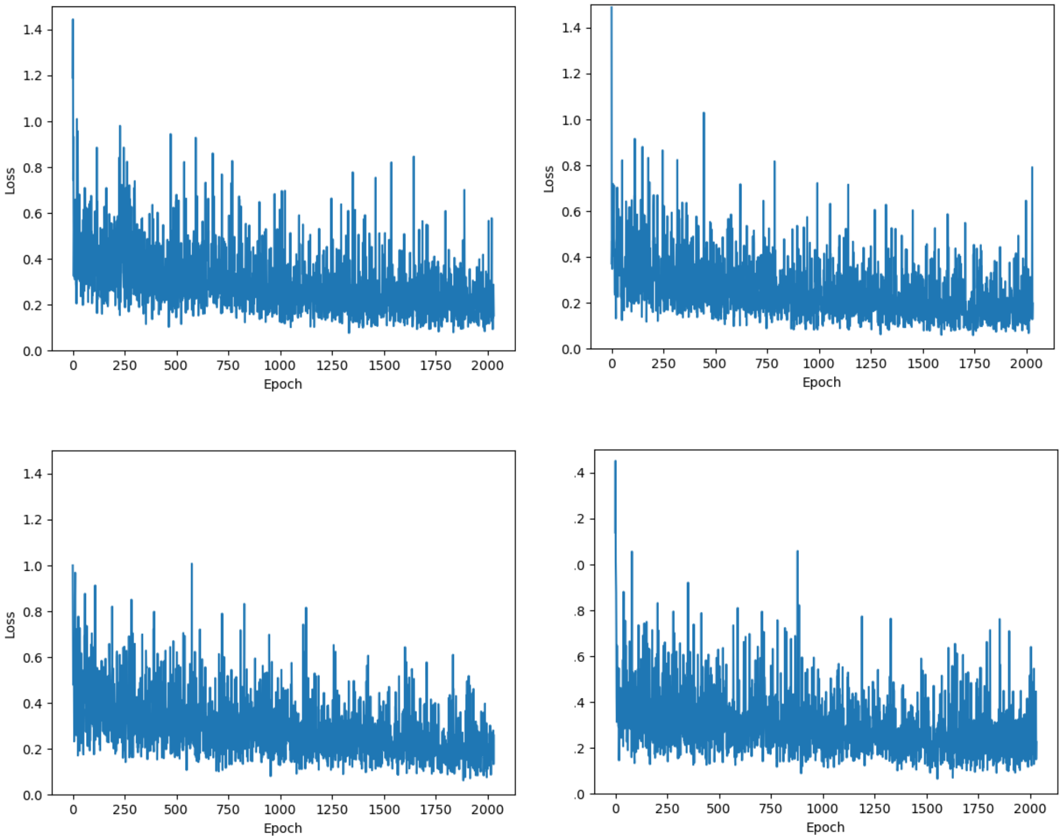


FIGURE 7 Loss performance metrics for 4 models. Top left: Regular Model. Top right: Shallow model. Bottom Left: Model with altered filters. Bottom Right: Deeper model with 5 fully connected layers and altered filters.

5.2.2 Accuracy

Due to the nature of that this task, the confusion matrix is generated based on bin sizes. Unlike a classification task, even if the prediction lies outside the ground truth, there is still some usefulness of the result if the results lie in an adjacent bin; In the confusion matrix shown in Figure 8, for occupancy, for example, a prediction of an occupancy of 31% coverage would be considered a misclassification if the ground truth is 29% coverage, based on how the bins are arranged. A list of accuracies in relation to different bin sizes are presented in Table 4.

Table 4: Model Accuracy based on different bin size

Bin Size (% Coverage)	Accuracy (%)

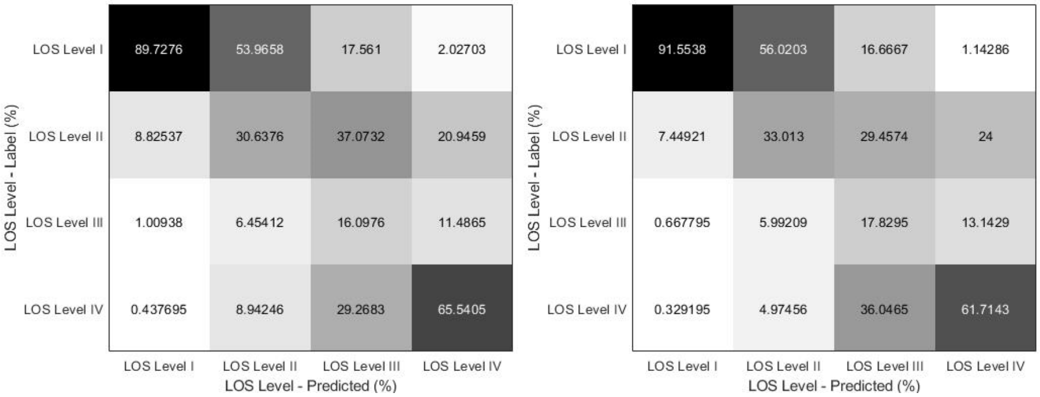
2.5	23.6
5	41.1
7.5	54.8
10	72.4
12.5	83.0
15	88.6

5.2.3 Model comparisons

The metric for classifying these CNNs is the Level of Service (LOS) index, an idea that segments density into different levels. The LOS values have been modified from veh/km/lane to percent coverage due to the lack of information regarding the number of lanes, as well as assumption of correlation between percent coverage and density. Due to the fact that the accuracy is dependent on bin size as described above, we adapted the LOS metric, which is then used to determine the most suitable architecture. The models all have similar classifications, with the exception of the deep model, which had significantly worse classification accuracy for LOS levels III and IV.

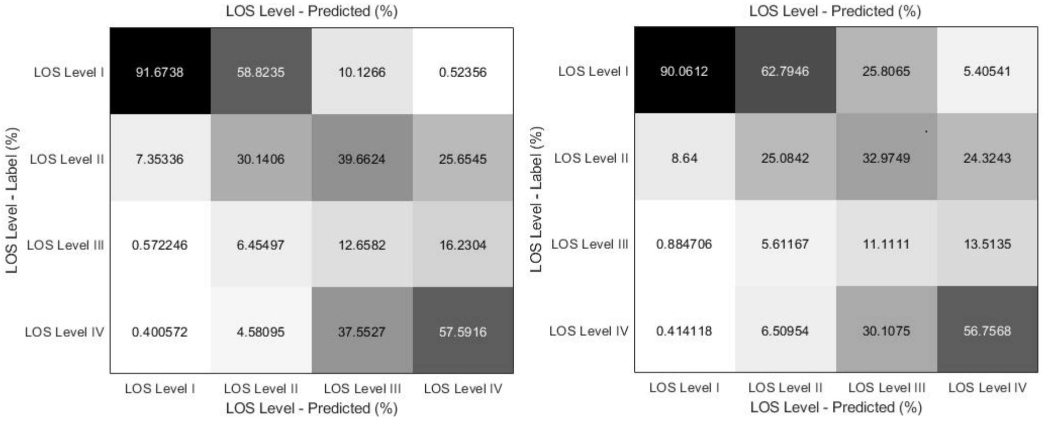
Confusion matrices for the 13,248 images in the validation set were generated.Note that the inputs and outputs are floats; this means that a typical confusion matrix for classification will not work, and therefore an adapted confusion matrix must be used, where values are binned. The network outputs a fairly accurate prediction for values in this range.

It can be seen that all models have the most trouble classifying LOS levels between II and III, and perform better for LOS levels of I and IV. This would make sense, as flow characteristics for a medium amount of traffic is expected to be harder to distinguish visually compared to both extremes. There is a strong tendency to classify LOS level II as I and LOS level III as II or IV, which may suggest visual similarity or lack of training data for these classes. The results are presented in Figure 8.



	LOS Level I	LOS Level II	LOS Level III	LOS Level IV
LOS Level I	89.7276	53.9658	17.561	2.02703
LOS Level II	8.82537	30.6376	37.0732	20.9459
LOS Level III	1.00938	6.45412	16.0976	11.4865
LOS Level IV	0.437695	8.94246	29.2683	65.5405

	LOS Level I	LOS Level II	LOS Level III	LOS Level IV
LOS Level I	91.5538	56.0203	16.6667	1.14286
LOS Level II	7.44921	33.013	29.4574	24
LOS Level III	0.667795	5.99209	17.8295	13.1429
LOS Level IV	0.329195	4.97456	36.0465	61.7143



	LOS Level I	LOS Level II	LOS Level III	LOS Level IV
LOS Level I	91.6738	58.8235	10.1266	0.52356
LOS Level II	7.35336	30.1406	39.6624	25.6545
LOS Level III	0.572246	6.45497	12.6582	16.2304
LOS Level IV	0.400572	4.58095	37.5527	57.5916

	LOS Level I	LOS Level II	LOS Level III	LOS Level IV
LOS Level I	90.0612	62.7946	25.8065	5.40541
LOS Level II	8.64	25.0842	32.9749	24.3243
LOS Level III	0.884706	5.61167	11.1111	13.5135
LOS Level IV	0.414118	6.50954	30.1075	56.7568

LOS Level - Predicted (%)
LOS Level - Label (%)
LOS Level - Predicted (%)
LOS Level - Predicted (%)
LOS Level - Label (%)
LOS Level - Predicted (%)

FIGURE 8 LOS confusion matrices for alternative models. Top left: Original Model. Top right: Shallow model. Bottom Left: Model with altered filters. Bottom Right: Deeper model with 5 fully connected layers and altered filters.

5.2.4 Issues

We attempt to determine different flow regimes based on speed and occupancy data. Speed and occupancy information is found using the associated loop detector values for each image; a total of 35,000 points were collected. Due to the low dimensionality, trends can be seen through inspection. More advanced clustering methods are not required.

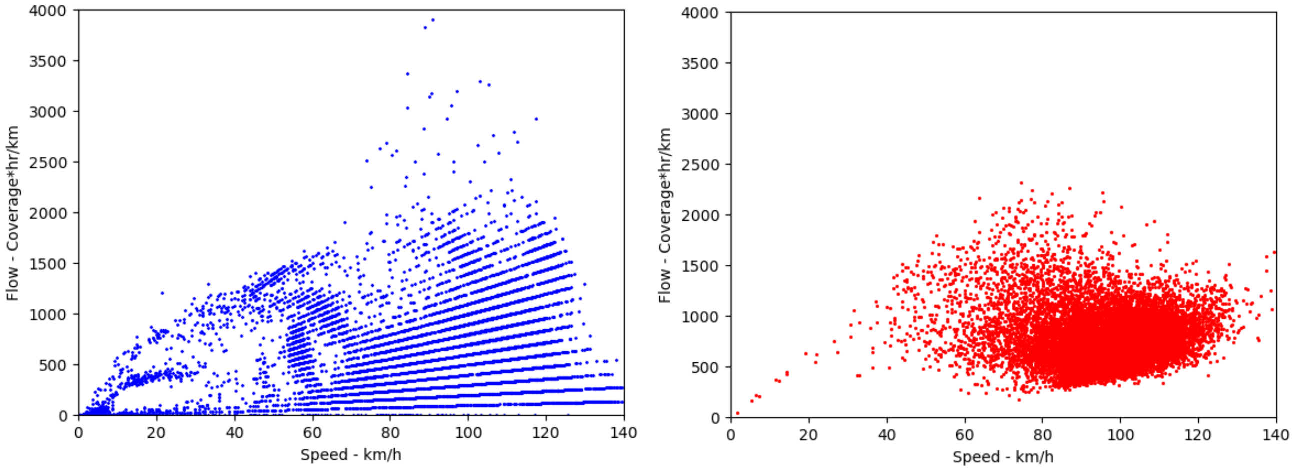


FIGURE 9 Plot of speed vs flow for ground truth (left) vs the predicted value (right).

The model shows a trend between speed and occupancy, although it does not appear to predict a decreased occupancy for the congested regime. In addition, it can be seen that the model does not predict the long tail where the occupancy stays constant from speeds of 100 to 140km/h - this is to be expected, as there is no significant visual difference between free-flow traffic at 100km/h as opposed to 140km/h.

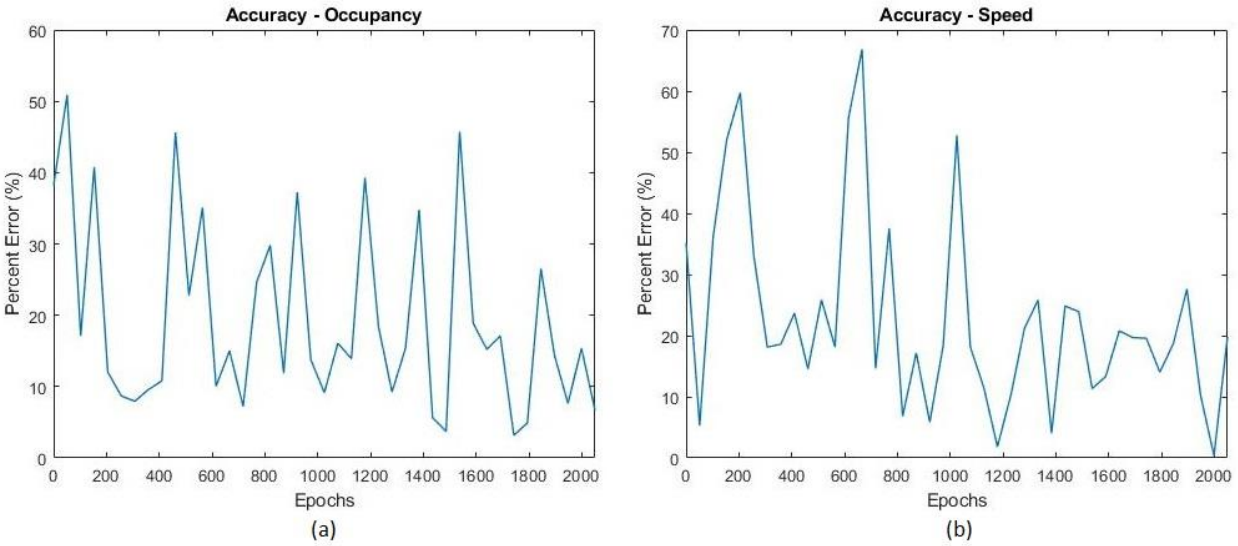


FIGURE 10 Accuracy. (b) Accuracy - Occupancy (c) Accuracy – Speed

The above results show that the model requires further improvements with classifying traffic in the high-occupancy regime. Figure 11 is a good example of a batch with a high number of reference pictures with a high occupancy; it can be seen that the model consistently predicts a higher occupancy for the images with a high occupancy, but falls short. This is likely due to the imbalance of images with low congestion and images with high congestion, as can be seen in Figure 6. It is worthy mentioioning that the loop detector sometimes fails to detect the speed and consequently outputs a abnormally low speed; this is more of a consequence of faulty loop sensors in the training data.

As a result, the error sometimes increases, if the batch has a high number of images with a high occupancy rate, this could increase the batch relative error. This could be rectified by adding more training images with a high occupancy to address the class imbalance, for future improvements to the model. Further misclassification could be attributed to failures on the part of the vehicle segmentation algorithm, which fails for times of day with rapidly changing lighting conditions. Shadows are not accounted for by the algorithm and could negatively affect classification. Some failure cases are shown in Figure10. Further work will be done to identify and remove shadows.

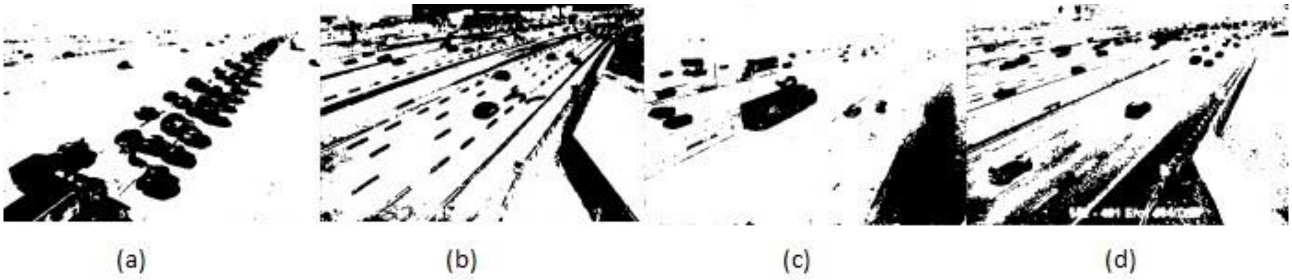
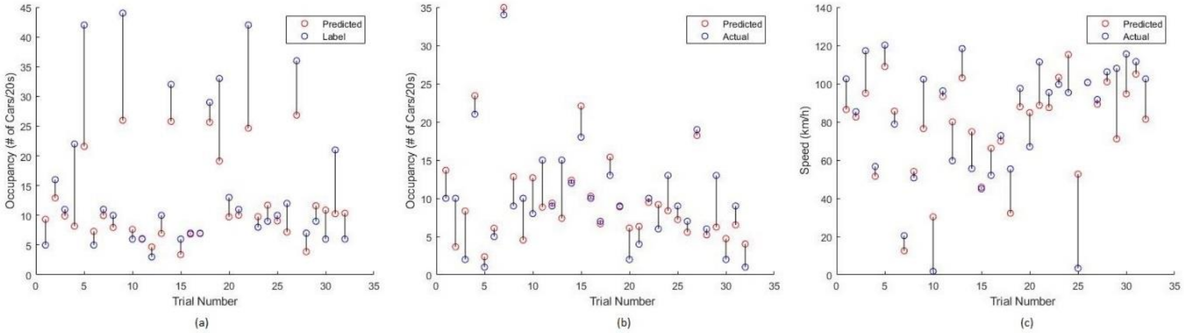


FIGURE 11 Failure cases for segmentation algorithm. (a) A low sun angle causes shadows to be binarized in the segmentation algorithm. (b, d) A low sun angle changes the light reflected by reflective surfaces, such as concrete and lane markings, quickly. (c) Grass textures are not properly segmented. A low sun angle causes the asphalt itself to become reflective at certain times of day.



(a)

(b)

(c)

FIGURE 12 Some examples of predicted vs. actual results. (a) demonstrates misclassification in the congested regime; there is a bias towards higher occupancy predictions, but the model often underestimates occupancy. (b) demonstrates a typical case for occupancy prediction, while the right (c) is a typical case for speed prediction. Notice the tendency for misclassification for slow flowing traffic.

6.0 CONCLUSION

We have developed a model for the classification of traffic characteristics based on traffic images, with the ground truth being values outputted by the loop detectors. This model shows some promise for the classification of traffic in the high and low occupancy regimes, although it struggles with classification in between the two regimes. According to the LOS metric defined in 6, the model has higher classification accuracy for LOS levels I and IV, with a wider spread at LOS levels II and III. Some parameters, such as accuracy and loss, were not as indicative of model accuracy, as the model is not doing classification work, and a predicted float that varies significantly from the labelled float can heavily skew these metrics.

This model has the ability to determine traffic characteristics using existing infrastructure that is in place, such as the Greater Toronto Area (GTA). This adds value by giving cities an alternative tool for congestion monitoring essentially for free, and can be an added feature in any traffic classification tasks using an ensemble approach. An approach similar to our confusion matrices generated for our modified LOS index could be used, with traffic flow binned at 15% road coverage; this bin size is reasonable for implementing a control system for traffic monitoring; having too high of a sensitivity would lead to instabilities caused by fluctuations by the control system, be it ramp metering or VACs technologies.

Low-level preprocessing of lane detection could be useful for supplementing another set of features that is unused by the network; directional segmentation would result in the ability to match cameras with the appropriate loop detector, as opposed to simply observing the state of traffic on a given road link (4). This would further improve our model. The principal emphasis areas remain theory and modelling of traffic flow interactions in a connected and automated system and the supporting data collection and mining efforts needed for calibration and validation purposes.

Better training data with lane labelling and directionality identification is expected to improve performance of this model. Due to the threshold method used, shadow detection and reduction is not implemented. This poses as an issue for low sun angles, as the shadow is not factored in by the preprocessing. The training data provided is limited to between 7am and 8pm to prevent shadows from posing a significant issue.

Future work will be done in detecting bridges, lanes and other features that help determine the region of interest, as opposed to the entire image; this will then be able to differentiate between different directions, as well as differentiate between the road of interest and surrounding roads, ramps and bridges. In addition, work will be done for bidirectional direction, where the loop detectors will be correlate with the appropriate direction, as well as the appropriate lanes.

Author Contribution Statement

The authors confirm contribution to the paper as follows: study conception and design: A. Aqra; data collection: Z. Ge; analysis and interpretation of results: A. Aqra , Z. Ge; draft manuscript preparation: Z. Ge, A. Aqra. All authors reviewed the results and approved the final version of the manuscript.

References

1. Shanghang Zhang, Guanhuang Wu, JoÃ£o Costeira, JosÃ© Moura. *Understanding Traffic Density from Large-Scale Web Camera Data.* Carnegie Mellon University, Pittsburgh, PA, USA, ISR-IST, Universidade de Lisboa, Lisboa, Portugal.
2. Rowa'a Jamal, Karmel Manaa, Maram Rabee'a and Loay Khalaf. *Traffic Control by Digital Imaging Cameras.* Electrical Engineering Department, University of Jordan, Amman, Jordan.
3. Alfredo Nantes, Dong Ngoduy, Ashish Bhaskar, Marc Miska and Edward Chung. *Real-time Traffic State Estimation in Urban Corridors from Heterogeneous Data.* Smart Transport Resarch Centre, Queensland University of Technology, Brisbane, Australia.
4. Jing Chen, Evan Tan and Zhidong Li. *A Machine Learning Framework for Real-Time Traffic Density Detection*. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 23, No. 7. 2009.

5. Chaochao Fan, Jun Tang, Nian Wang and Dong Liang. *Rich Convolutional Features Fusion for Crowd Counting.* School of Electronic Information Engineering, Anhui University, Hefei, China.
6. Xiaoyu Liu, Denning Campbell and Zhengyu Guo. *Single Image Density Map Estimation Based on Multi-Column CNN and Boosting.* School of Computing Science, Simon Fraser University.
7. Vishwanath Sindagi and Vishal Patel. *CNN-Based Cascaded Multi-task Learning of High-level Prior and Density Estimation for Crowd Counting.* Department of Electrical and Computer Engineering, Rutgers University.
8. Jiyong Chung and Keemin Sohn. *Image-Based Learning to Measure Traffic Density Using a Deep Convolutional Neural Network*. IEEE Transactions on Intelligent Transportation Systems, Vol. 19, No. 5, May 2018.
9. Gerdien A. Klunder, Henk Taale, Leon Kester and Serge Hoogendoorn. *The Effect of Inaccurate Traffic Data for Ramp Metering: Comparing Loop Detectors and Cameras using Information Utility.* Delft University of Technology, TNO, Smart Mobility.
10. Yi-Chang Chiu, Jon Bottom, Michael Mahut, Alex Paz, Ramachandran Balakrishna, Travis Waller and Jim Hicks. *Dynamic Traffic Assignment.* Transportation Research Board, 500 Fifth Street, NW, Washington, DC 20001.
11. Mohamed Elshenawy, Mohamed El-darieby and Baher Abdulhai. *Automatic Omputation of Missing Highway Traffic Volume Data.* University of Toronto, Canada, University of Regina, Canada.
12. V. Thamizh Arasan, G. Dhivya. *Measurement of Occupancy of Heterogenous Traffic Using Simulation Technique.* Transport Engineering Division, Department of Civil Engineering, IIT Madras, Chennai.