IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 5, 2019, accepted November 19, 2019, date of publication November 22, 2019,
date of current version December 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955307

Image-Based Learning to Measure the Stopped Delay in an Approach of a Signalized Intersection

JOHYUN SHIN, SEUNGBIN ROH, AND KEEMIN SOHN
Department of Urban Engineering, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Keemin Sohn (kmsohn@cau.ac.kr)

This work was supported in part by the Chung-Ang University Research Scholarship Grants in 2018, and by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport under Grant 19TLRP-B148659-02.

ABSTRACT Traffic delays are inevitable when evaluating the performance of a signalized intersection, but these delays cannot be directly measured in the field based on existing spot detectors. Traffic-light controllers have adopted a reinforcement learning (RL) algorithm, which is currently prevalent in the field of study and requires real-time measurement of traffic delays to derive the state and reward for each time period. No RL-based study, however, has provided a robust way to measure traffic delays. In order to bridge the gap, we devised a convolutional neural network (CNN) to directly measure traffic delays from video footage in an end-to-end manner. The proposed methodology proved superior to both a state-of-the-art vision technology and an analytic formula that has widely been used to estimate delays. Furthermore, a robust method to secure labeled data without human input was suggested based on a cycle-consistent adversarial network (CycleGAN).

INDEX TERMS Traffic delay estimation, image-based learning, deep convolutional neural network.

I. INTRODUCTION

Traffic delays have been used as a basic indicator to assess the performance of traffic light control algorithms. Many traffic signal controllers depend on traffic delays to obtain an optimal signal phase plan. The highway capacity manual (HCM) provides an analytic formula to estimate traffic delays for a lane group at a signalized intersection [1]. However, it is well known that the formula does not work well when traffic flows are saturated or where cars arrive on an irregular basis.

Traffic-light controllers have adopted a reinforcement learning (RL) algorithm, which is currently prevalent in the field of study and requires real-time measurement of traffic delays to derive the state and reward for each time period. Most researchers who have studied RL-based traffic control have assumed that vehicle delays could be easily measured [2]–[5], but the reality is that no real-time measurement system for delays is available in the field. If every vehicle could be embedded with an on-board unit it might be possible to transmit their mobile information for every short period of time to compute traffic delays. However, such a fully connected environment cannot be realized any time soon.

It is necessary to find a robust way to measure traffic delays in an intersection approach on a real-time basis. The present study was focused on the possibility that a traffic delay could be measured from video images using deep-learning technology that has recently produced breakthroughs [6]–[10]. Some pioneers have already attempted to use image analysis to automate the estimation of stopped delays at signalized intersections [11], although they were forced to depend on rule-based engineering rather than on data-driven learning methodologies.

A deep convolutional neural network (CNN) has recently been used to measure both the traffic density and the space mean speed of an intersection approach based solely on video images [12], [13]. These previous studies have shed light on the possibility that traffic delays could also be measured from images. A CNN for measuring the space mean speed was adapted to differentiate the number of stopped vehicles from all other moving vehicles at an intersection approach. This counting was implemented by adopting two consecutive photos taken over a short period of time to be used as input. Once the number of stopped vehicles is measured, it is easy

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

to compute the total cumulative stopped delay. Theoretically, the total delay can be obtained by integrating the number of stopped vehicles (= queue length) over time. This two-step approach was devised in the present study to measure traffic delays.

The institute of transportation engineers (ITE) recommended that a manual delay measurement method be used in the field [14]. For 15 minutes, the number of stopped vehicles in an intersection approach is counted every 15 seconds, and the number is multiplied by 15 seconds and summed over a 15-minute time period. The average delay can be obtained by dividing the total delay by the number of vehicles passing the stop line during the 15-minute time period. This method has a drawback wherein a 15-second time interval can be too long to accurately convey the dynamics of vehicles. That is, the assumption that vehicles maintain their behavior during the 15 seconds is so naïve that the computed delay cannot secure an acceptable level of accuracy.

A novel technology was devised in the present study to overcome the problem. A CNN continuously counted the number of stopped vehicles within a much shorter time interval. Two consecutive video images were used as input for a CNN to count the number of stopped vehicles. If two consecutive images are given, it is easy for a human to recognize whether vehicles within the images are moving. Of course, this judgment could be automated based on an engineering-based method developed in a previous study [11]. On the other hand, the purpose of the present study was to develop an end-to-end learning model that could be used to count the number of stopped vehicles without any feature engineering or any rule for judgment. An end-to-end CNN model was trained only by image data to count the number of stopped vehicles. Counting the number of stopped vehicles in a short time interval led to an accurate measurement of the stopped delay. To the best of our knowledge, such a data-driven approach to measure traffic delays has never been attempted before.

The only concern with respect to the proposed approach is the difficulty in securing labeled images to train the proposed CNN. It is a formidable task to manually count the stopped vehicles for a large image set. To circumvent this burdensome task, a revolutionary method was adopted using a generative adversarial framework. Our previous study [13] already succeeded in synthesizing traffic images from naïve animation images created by a traffic simulator, which had perfect traffic information that could be used as a label when training a CNN model to measure the space mean speed. The method was also adopted in the present study to synthesize real-looking images tagged with the number of stopped vehicles.

Another approach to counting stopped vehicles on a road segment involves detecting and tracking individual vehicles in video frames. Although there is no learning-based model to directly measure traffic delays, object-detection technology with a filtering algorithm could count the stopped vehicles from video footages. A YOLO-based vehicle detection-and-tracking model was chosen as a reference [15], [16], and its performance in counting stopped vehicles was compared with that of the proposed method. The merits and drawbacks of the reference model were fully discussed in the present study.

The present paper is structured as follows. The next section provides a literature review for previous efforts to estimate traffic delays including an effort to develop analytic formulas. The third section describes how to collect image data for training the proposed model. How to setup a CNN architecture to count the stopped vehicles is introduced in the fourth section. The fifth section presents the test results of the proposed model trained on manually labeled images and shows the traffic delay can be measured without errors. The model performance was compared with two different contexts in the sixth section. Traffic delays from the proposed model were compared with those from a conventional analytic formula and those from a detection-and-tracking algorithm. In the seventh section, how to create training images without human effort is described, and the performance of the model trained on synthesized images is compared with that of the model trained on real images. In the last section, conclusions are drawn, and further studies are proposed to substantiate the present study.

II. RELATED WORK

Traffic delays at intersections have been estimated in an analytic manner using formulas. Most studies adopt analytic formulas to derive the average delay in an intersection approach within a certain time period (e.g., 15 minutes). According to Cheng *et al.* [17], theoretical delay models have been developed through three stages. The first stage was prevalent in the 1920s-1970s, wherein researchers assumed a steady state to estimate traffic delays. Models succeeded in estimating delays only for under-saturated traffic conditions, whereas they failed to obtain a reliable estimation when an intersection approach was congested.

In the second stage (1970s-2000s), researchers adopted the same analytic formula used to estimate an average delay, but they employed adjustment factors to consider various vehicle arrival patterns from an upstream intersection. Their model performance was somewhat enhanced, but it retained the intrinsic drawback whereby dynamic changes in traffic delays could not be incorporated into the formula. The third stage followed the new millennium, and some modifications were made for the existing formulas in order to resolve miscellaneous problems that had been posed in the second stage. Meanwhile, new technologies such as fuzzy logic algorithms, image analytics, and artificial neural networks were introduced to estimate traffic delays.

Qiao *et al.* [18] developed a fuzzy logic-based model to estimate intersection delays. They adopted new variables for ambient conditions that included parking, the stopping of busses, visibility, and climate, as well as employing basic variables such as the degree of saturation, cycle length, and green time ratios. They argued that their model performance far surpassed that of the existing analytic formulas.

As mentioned earlier, image analytics was also employed to estimate the delay in an intersection approach [11]. A virtual sensor in the path of vehicles was placed on images and kept track of the digital signals of the vehicles and their locations to judge whether vehicles were moving. This measurement system depended on several engineering judgments to recognize the presence and movement of vehicles. In fact, the approach did not adopt a learning-based model but employed an engineering-based methodology. This means that the approach could not be improved regardless of how much image data were provided.

Murat and Baskin [19] used an artificial neural network to measure non-uniform delays that could scarcely be estimated using existing analytic methodologies. They chose various forms of input data to feed the neural network but did not recognize the utility of images as input for estimating traffic delays, because a deep learning model had not yet received proper consideration at the time their study was being conducted. All those methodologies are outdated with arrival of the era of deep-learning and depended largely on independent variables that cannot be easily acquired in the field.

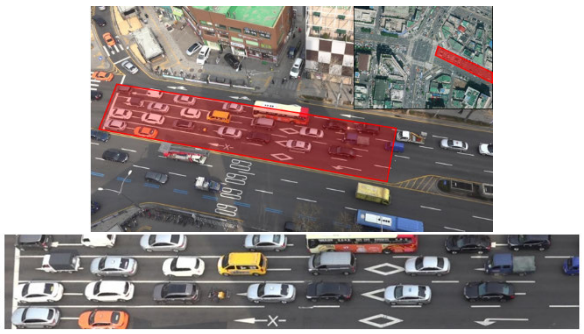
With the advance of deep learning technologies, there have been many attempts to count vehicles using traffic images. Almost all previous attempts adopted a two-step procedure: i.e., detecting and tracking individual vehicles in consecutive video frames. As a reference approach to estimate traffic delays, the potential of this technique to discern stopped vehicles from moving ones was the focus. Deshmukh and Uke (2016) summarized the current prevailing methodologies used to detect and track moving vehicles based on vision technologies [20]. Background subtraction methods have been the most commonly employed techniques to detect vehicles within an image [21]–[23]. Subtracting the current image from a background image transforms an original image into a silhouette with blobs representing vehicles. A Gaussian mixture model (GMM) has also been widely used to determine whether pixels correspond to background or foreground in a probabilistic manner [24], [25]. An optical flow method is another robust method used to detect and track moving vehicles [26]. For optical flow, the change in pixel color is a key to derive a two-dimensional velocity vector to trace a vehicle's motion.

More recently, the performance of YOLO has been highlighted in vehicle detection studies [27], [28]. Some researchers developed vehicle counters that incorporate YOLO and a filtering algorithm [15], [16]. Once a YOLO detects vehicles in each video frame, a filtering algorithm tracks them across many video frames. The present study showed that the YOLO-based counter could be revised to count the number of stopped vehicles. The proposed method will be compared with a YOLO-based model in section VI.

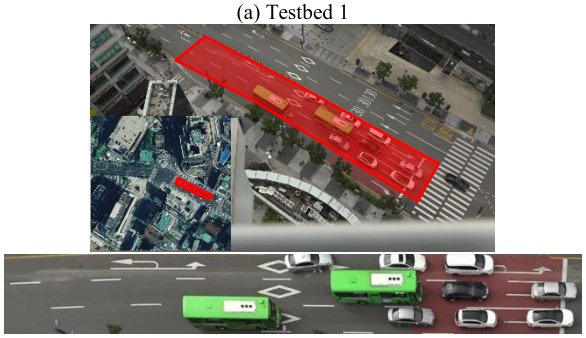
As shown in the review above, the effort to measure vehicle delay has changed from one of using an analytic estimation with a formula into adopting a data-driven approach via artificial intelligence. Based on the gains of previous studies, the present study benefitted from the insight of combining the utility of a neural network with the potential power of input from images, and the existing notion that was confined to the analytic formula was shifted to the direct measurement of delays in the field. It is apparent that learning-based image sensors will soon replace engineering-based sensors in traffic surveillance. The present study took the initial step to accomplish this new direction.

III. DATA COLLECTION

Video images were taken at two different intersection approaches located in Seoul, Korea for 98 minutes in daytime on a weekday. The first testbed was 67 m long and 13 m wide with 4 lanes, and the second one was 59 m long and 10 m wide with 3 lanes (see Fig. 1). For the first testbed, the upper two lanes were reserved for straight-ahead traffic, and the lower two lanes for left-turn traffic. For the second testbed, the upper two lanes were reserved for straight-ahead traffic, and the remaining lane for left-turn traffic. The test image was cropped from a larger photo that covered the entire testbed. Such birds-eye view photos are available owing to CCTVs for traffic accident surveillance. The stop line was located at the left (or right) end of the testbed. The traffic signal control parameters such as cycle length (= 140 sec for the first testbed and 160 sec for the second testbed) and green time ratios were collected in the field.



(a) Testbed 1



(b) Testbed 2

FIGURE 1. Testbeds for the present study to measure traffic delays. For each testbed, the upper photo is a raw video shoot taken from 35 to 45 m high, and is similar to a photo that could be taken by a CCTV in the area. A small image inset in the upper right (or lower left) corner of the photo shows a higher aerial view and is provided to promote a better understanding of the entire intersection. For each testbed, the lower photo was clipped from the upper photo and the viewport and angle were adjusted to provide rectangle input for the proposed CNN model.

For the video frames taken for 98 minutes, the number of stopped vehicles were recorded manually every 0.2 seconds. The pseudo-ground truth for delay was computed using this short time interval and was used to assess the performance of the proposed model. On the other hand, 10,000 pairs of consecutive video images were chosen for each testbed to train the proposed CNN model to recognize the number of stopped vehicles. Another set of image data (= 2,000 pairs of images) was chosen from the remaining images to test the trained model for each testbed. For the time interval of 1 second, only 5,880 pairs of images were available. To train the model, 4,500 pairs of images were used, and the rest of the images were reserved for the test. While training the model, 10% of training images were randomly chosen and used for the cross-validation.

In fact, labeling images was a burdensome task for the modeling to count the number of stopped vehicles. Nonetheless, in the present study all images (29,400 images × 2 testbeds) for 98 minutes were manually labeled to verify whether the proposed approach could continuously measure the cumulative delay.

To replace the manually tagged images, fake photos that appeared real were synthesized by the coordination of both a traffic simulator and a cycle-consistent generative adversarial network (CycleGAN) [13]. Initially, the physical layout and operational conditions of the first testbed were fed to a commercial traffic simulator (Vissim). The simulation run was then tuned so that the simulation environment could be the same as the real traffic conditions. The simulation run offered cartoon-like animation images that cannot be directly used to train a CNN to measure traffic parameters.

A CycleGAN was trained on two unpaired sets of real photos and animated images. The trained CycleGAN synthesized the real-looking fake photos from the same number of randomly chosen animation images (= 10,000 pairs of images). The synthesized photos had a perfect label (= the number of stopped vehicles), since the original animation images were generated from controlled traffic simulation. A sample of the animated and synthesized photos for the first testbed will be shown in the seventh section.

IV. MODEL ARCHITECTURE

The delay measurement becomes a trivial task once the number of stopped vehicles can be counted continuously over time. Eq. (1) indicates the theoretical background for computing delays. The cumulative traffic delay can be obtained by integrating the instantaneous count of stopped vehicles over time, as shown in Eq. (1).

$$D(T) = \int_{t_0}^{T} Q(t)\, dt \tag{1}$$

In Eq. (1), $D(T)$ is the cumulative delay from the initial time, t_0, to a certain time, T, and $Q(t)$ is the number of stopped vehicles instantaneously observed at time t.

As a practical matter, the mathematical integration can be numerically approximated as in Eq. (2). Regarding the approximation, it is important to reduce the time interval at which the number of stopped vehicles is counted, as shown in Eq. (2).

$$D(T) \cong \sum_{i=0}^{(T-t_0)/\Delta t} Q'(i)\Delta t \tag{2}$$

In Eq. (2), Δt is a time interval to discretely measure the number of stopped vehicles and $Q'(i)$ is the number of stopped vehicles during the i^{th} interval [i.e., $Q'(i) = \mathrm{Q}(t_0 + \mathrm{i} \cdot \Delta t)$].

As mentioned earlier, the ITE recommended that the time interval should be 15 seconds when the delay is measured manually in the field [14]. Although this works well for the purpose of evaluating the intersection service at the macro level, it is impossible for the resultant delay measurement to be applied to an adaptive RL traffic signal control on a real-time basis.

The proposed CNN counted the number of stopped vehicles every 0.2 seconds, which made it possible to continuously measure the cumulative delay. As a human recognizes stopped vehicles using video frames, the proposed CNN counted the number of stopped vehicles based solely on two consecutive video images. This end-to-end manner has the great advantage of requiring no engineering judgment in the model. Once a sufficient amount of labeled images is provided for training, the CNN carries out the remaining tasks necessary to count the number of stopped vehicles.

The proposed CNN architecture was devised with multiple outputs such that the number of stopped vehicles can be counted for each lane group and each vehicle type (see Fig. 2). This was very efficient since a single neural network can accommodate multiple tasks. As a result, the proposed CNN simultaneously counted and classified stopped vehicles for each lane group. More concretely, the CNN was designed such that vehicles for two different lane groups could be separately counted within a single model [see Fig. 2]. No prior remedy for the input images was necessary to make the model recognize each lane group. A pair of full images covering both lane groups was used as input, and the CNN automatically classified straight-ahead and left-turn traffic. While training the CNN, the output nodes were fed with the numbers of stopped vehicles in the straight-ahead and left-turn lane groups, respectively.

The detailed CNN architecture was chosen on a trial-and-error basis. The previous CNN for measuring the space mean speed was adopted as a baseline model [13] and was adapted to recognize stopped vehicles after testing as many alternatives as possible. More concretely, hyper-parameters to determine the model architecture included the number of layers, number of nodes in each layer, number of filters for each layer, filter size, and stride size. The architecture was selected after testing 50 different combinations of hyper-parameters, each of which was randomly chosen within predefined ranges. According to Bergstra and Bengio [30], random searches of 8 trials matched or outperformed the grid searches of (on average) 100 trials. Recently,

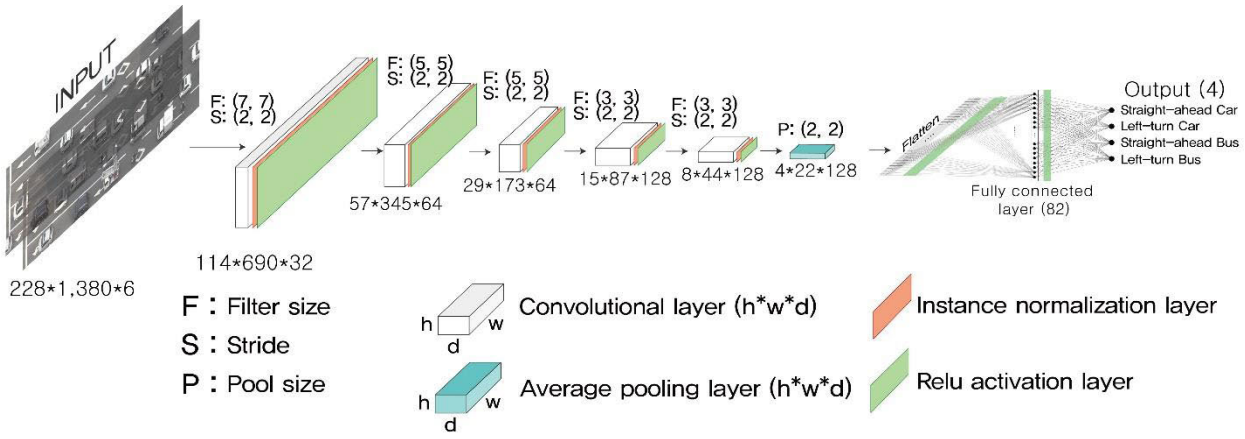


FIGURE 2. CNN architecture used to count the number of stopped vehicles. The input size is for the first testbed, but different input sizes are applicable with the filter layout fixed.

Bayesian optimization was applied to finding optimal hyperparameters of deep neural networks [31], [32]. This methodology would enhance the model performance in further studies.

The final architecture of the CNN model appears in Fig. 2. The CNN architecture was built by stacking 5 convolutional blocks, each of which was composed of a convolutional layer, an instance normalization layer, and a rectified linear unit (Relu) activation layer. Normalizing input feature maps on a layer-by-layer basis enhanced the performance of a deep neural network to abstract features from images. Instance normalization was used to independently compute the mean and standard deviation across spatial dimensions of each feature map within a batch input [33], which differs from batch normalization that computes them across an entire batch of images [34]. After the final convolution block, a single average pooling layer was added. The final tensor was flattened to feed the last fully connected layer that connected the output layer. The output layer had 4 nodes, each of which corresponded to counts of stopped cars and buses for each lane group, respectively.

V. DELAY MEASUREMENT RESULTS

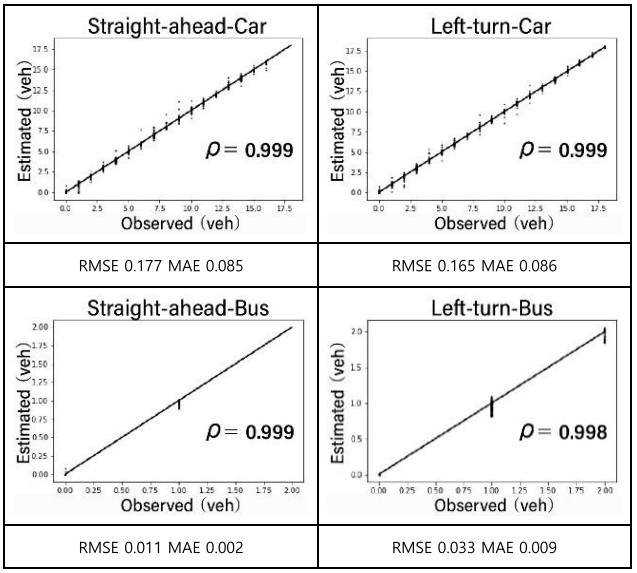
A. TEST RESULTS OF COUNTING STOPPED VEHICLES

The number of stopped vehicles was counted for both lane groups (i.e., straight-ahead and left-turn flows) prior to measuring the stopped delay. The number was also separately counted for two different vehicle types (i.e., cars and buses). The former encompasses cars, SUVs and small trucks, whereas the latter covers buses and large trucks. As mentioned earlier, when counting the number of stopped vehicles, an infinitesimally small interval is required to exactly measure the true delay. The pseudo-ground truth for delay was set in the present study based on the number of stopped vehicles counted every 0.2 seconds.

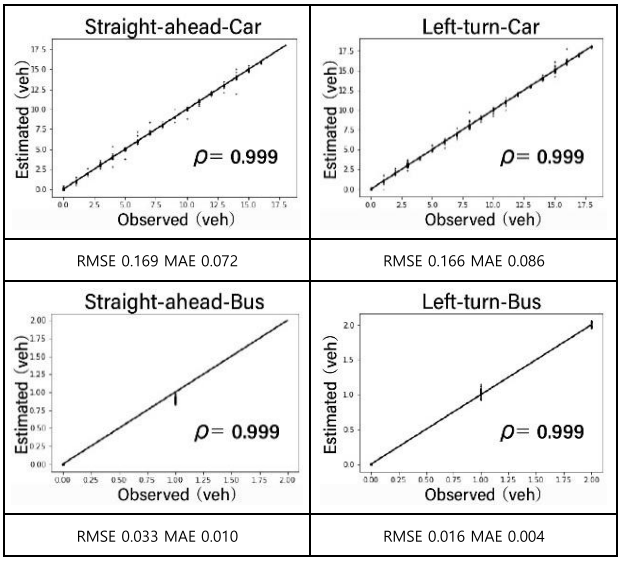
For each testbed, from among 29,400 pairs of consecutive images for a 98-minute survey period, 10,000 pairs of images were randomly selected for training the proposed model to count the number of stopped vehicles. Another 2,000 pairs of images were randomly chosen again for the test. The model performance derived from the test data are shown in this section. Three performance indices were adopted to confirm the utility of the proposed model [i.e., correlation coefficient (ρ), map absolute error (MAE), and root mean square error (RMSE).

For comparison, the same model architecture was trained on a different dataset collected from a longer time interval. A set of 5,880 pairs of images was manually labeled with the number of vehicles stopped during a 1-second interval. Fig. 3 shows performance indices of the proposed model for both time intervals of 0.2 and 1.0 seconds. For both testbeds, the model trained on images of 0.2-second intervals produced almost the same test results as the observed number of stopped vehicles. The 0.2-second interval proved to be almost perfect to count the number of stopped vehicles. When a larger time interval (= 1.0 second) was adopted, the discrepancy between the estimated and observed numbers of stopped vehicles was negligible, which means the labeling task in the future can be carried out more easily with a larger time interval [see Fig. 3 (b) and (d)].

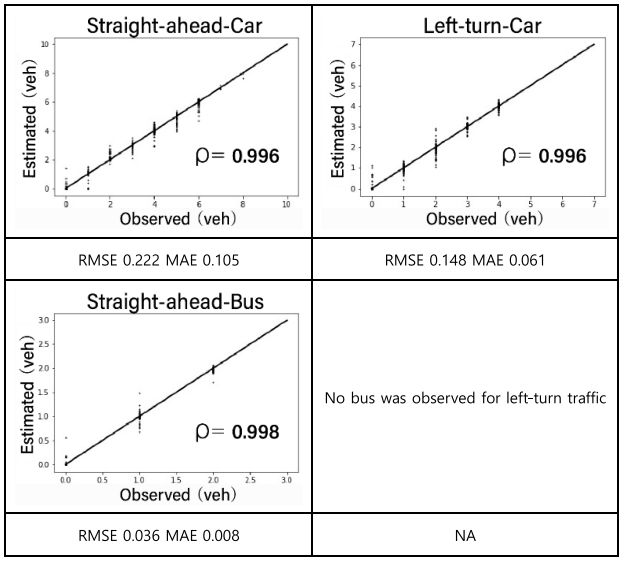
Deep learning models have been criticized due to an uncertainty wherein the model cannot account for causal effects between input and output [35]. Nonetheless, the results showed that CNN filters could learn to count the number of stopped vehicles based solely on two consecutive images. Although a direct interpretation is not possible, it is certain that weights for filters stored experiences while learning in order to improve performance. To show such evidence, values of the last 128 4×22 tensors of the proposed CNN are plotted in Fig. 4. for two contrast inputs: i.e., two consecutive images of running vehicles and those of stopped vehicles, respectively. The difference in filter values is apparent, and these different patterns made it possible to count stopped vehicles.



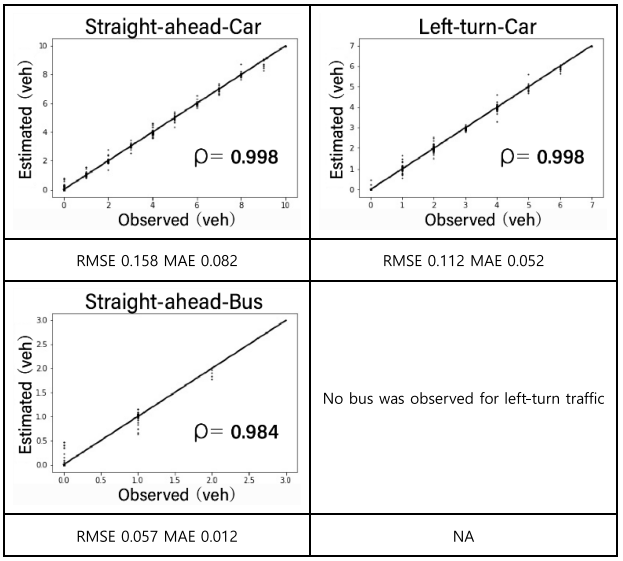
(a) Test result for testbed 1 (0.2 second interval)



(b) Test result for testbed 1 (1.0-second interval)



(c) Test result for testbed 2 (0.2-second interval)



(d) Test result for testbed 2 (1.0-second interval)

FIGURE 3. **Performance of the model trained on manually labeled images in counting the number of stopped vehicles.**

B. RESULTS FROM MEASURING THE STOPPED DELAY

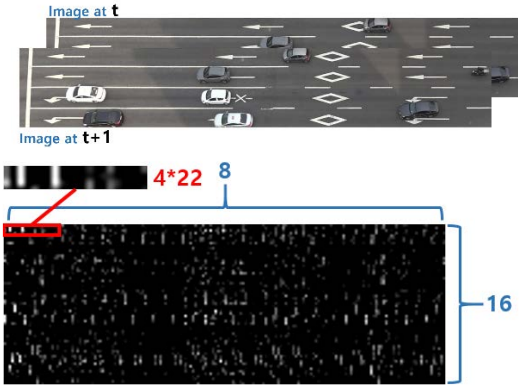
Counting the number of stopped vehicles was prerequisite to measuring the stopped delay. The stopped delay was estimated by accumulating the number of stopped vehicles counted in a short time interval. Fig. 5 shows the stopped delay for each lane group and for each vehicle type. When a 0.2 second time interval was employed, the estimated cumulative delay was almost the same as the pseudo-ground truth. As the time interval was increased to 1.0 second, the model performance deteriorated slightly only for straight-ahead buses for both testbeds. This could have been due to an insufficient amount of buses that kept stopping for 1.0 second, whereas there were more buses that kept stopping for 0.1 second. However, this was not a serious problem since the discrepancy was not significant.

VI. COMPARISON WITH OTHER METHODS

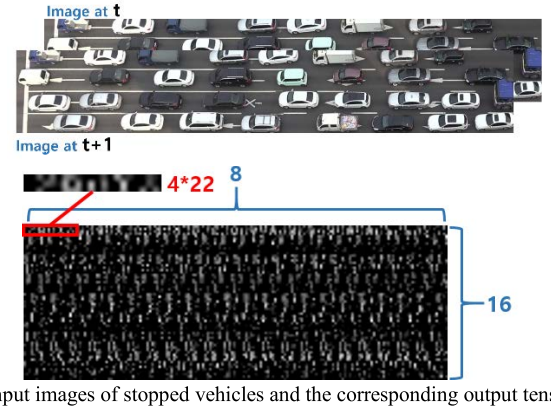
A. COMPARISON WITH AN ANALYTIC MODEL

The HCM 2000 presents a formula that can be used to estimate the average control delay for any time period [1]. The control delay includes the initial deceleration delay, the stopped delay, and the final acceleration delay. Unfortunately, the present approach can only measure the stopped delay. Thus, the delay formula from the previous HCM 1994 [36], which estimates the average stopped delay per vehicle for a 15-minute period, was adopted as a reference for comparison. Eq. (3) denotes the delay formula used in HCM 1994.

$$d_s = 0.38\frac{C\,(1-\lambda)^2}{\{1-\lambda\,[\min{(x,\,1.0)}]\}} + 173x^2\left[(x-1)+\sqrt{(x-1)^2+\frac{mx}{c}}\right] \tag{3}$$



(a) Input images of moving vehicles and the corresponding output tensors



(b) Input images of stopped vehicles and the corresponding output tensors

FIGURE 4. Values of the last 128 tensors of the proposed CNN for two contrastive inputs.

where,

d_s = stopped delay,

c = capacity of the lane group (vph),

x = v/c ratio for the lane group (= traffic volume/capacity),

λ = g/C ratio for the lane group (= green time/cycle length), and m = the incremental calibration term representing the effect of arrival type and the degree of platooning (the default value 16 was adopted in the present study).

For both testbeds, the signal parameters of both lane groups were surveyed in the field and applied to the formula to estimate the stopped delay. Six 15-minute periods out of 98 minutes were chosen to compare the measurement performance between the HCM formula and the proposed CNN models. Table 1 shows the comparison results. As expected, the proposed model outperformed the analytic formula. The average delays measured from the proposed model were much closer to the pseudo ground truth. The excellence lies in the fact that the proposed approach is purely data-driven, even though how the CNN model works so well is not yet fully accounted for.

For the first testbed, the HCM formula underestimated delays for a higher saturation level. This implies that the actual delay for a higher saturation level stems largely from unsteady traffic states rather than from uniform and incremental traffic states. On the other hand, the formula overestimated delays in the straight-ahead lane group of the first testbed for the period from 30 to 60 min when small traffic volumes were observed. In the same context for the second testbed, despite a longer cycle length and a shorter green time, much smaller traffic volumes in all time periods led the formula to overestimate traffic delays.

B. COMPARISON WITH AN EXAMPLE OF COMPUTER VISION TECHNOLOGY

As another reference, a vehicle detection-and-tracking model incorporating YOLO v3 and a SORT algorithm was selected [28]. The former was for vehicle detection, and the latter was for vehicle tracking. Since the existing YOLO failed in recognizing vehicles from a bird-eye view, we retrained it on a new dataset. We drew bounding boxes for all vehicles within 3,000 aerial road images. The tracking algorithm (SORT) was tuned to recognize whether a vehicle is moving. The reason a YOLO was selected as a reference was because the model has been a leader in the recent progression to deep learning and is known to be more promising in object detection than older vehicle detection models such as background subtraction or optical flow models [15], [27], [28].

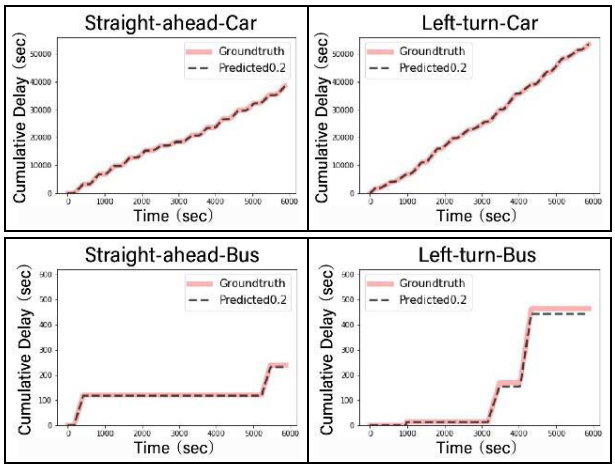
The performance of the YOLO-based model was compared with that of the proposed methodology and of the HCM formula. The YOLO-based model proved to be superior to the HCM formula but showed a comparable performance with our proposed methodology (see Table 2). However, drawing a bounding box for every vehicle in all training images was much more labor-intensive than counting the number of stopped vehicles to obtain labeled data. A revolutionary method to alleviate the effort to obtain labeled data was developed for the propose CNN, and is described in the next section.

The proposed CNN used to count the stopped vehicles had a much lighter structure than a YOLO model. The proposed approach to estimate delay also did not require an additional algorithm for vehicle tracking and thus requires much shorter computing time (see Table 2).

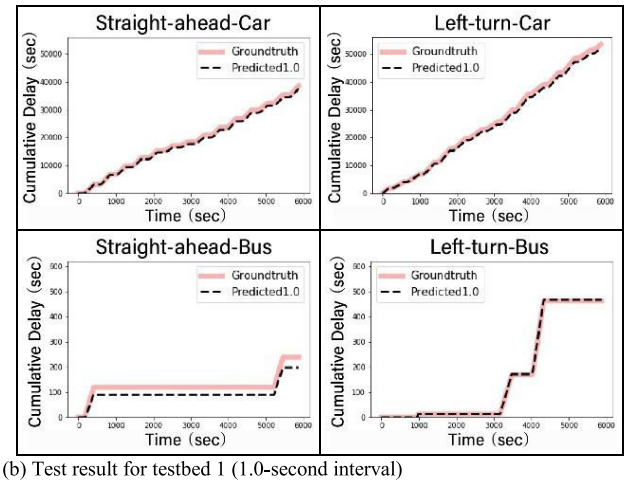
VII. ALLEVIATING THE EFFORT OF TAGGING IMAGES WITH LABELS

The only complication associated with the proposed method is the human effort that is required to tag images with a true label. A large number of labeled images (about 10,000 pairs of images) was necessary to train the proposed CNN. It took a couple of days for two proficient researchers to tag the images with labels. It is burdensome to manually tag images with true labels whenever a new intersection approach is included.

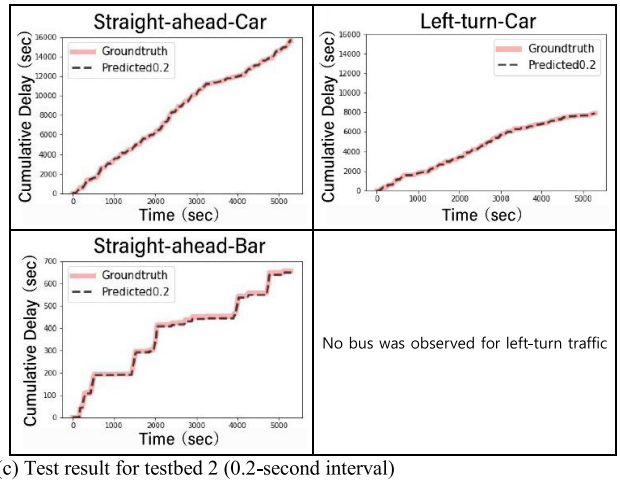
Our previous study provided a great possibility to overcome the difficulty in labeling. A CycleGAN was used to successfully synthesize virtual reality and create many labeled images with the help of a traffic simulator [13]. The methodology depended largely on a reliable traffic simulator that can mimic the traffic conditions in the testbed. It was verified in the first testbed that a CycleGAN could generate labeled images for pretraining a proposed model to count stopped vehicles.



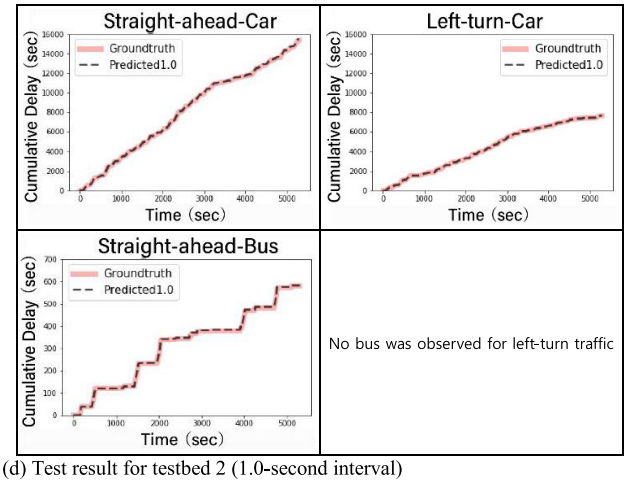
(a) Test result for testbed 1 (0.2-second interval)



(b) Test result for testbed 1 (1.0-second interval)



(c) Test result for testbed 2 (0.2-second interval)



(d) Test result for testbed 2 (1.0-second interval)

FIGURE 5. **Model performance in measuring the cumulative stopped delay.**

A commercial traffic simulator (Vissim) was used to replicate the real traffic conditions of the first testbed and then provided animation images for them. The images were, however, so naïve that they could not be directly used to train the proposed CNN model, although the images contained exact traffic information including the number of stopped vehicles. A CycleGAN converted these naïve animation images into seemingly realistic images to pretrain the proposed CNN. Both the generator and discriminator models were set up in reference to our previous study to measure the space mean speed. For the details of how to set up and train the models, readers are referred to the previous work [13].

Fig. 6 shows randomly chosen examples of the conversion. Images on the left were generated by the traffic simulator, and images on the right were synthesized from those on the left. The same number of synthesized images (= 10,000 pairs) was used to pretrain the CNN to maintain consistency with the model training based on real images with true labels.

The performance of the model pretrained on the synthesized images was tested on the same test data that were manually labeled (= 2,000 pairs). A time interval of 0.2 sec was applied to the comparison. Fig. 7 shows the model performance to count the number of stopped vehicles. The test results were compared with those from the CNN trained on manually labeled data (see Table 3). The performance of the model pretrained on synthesized images was inferior to that of the model trained on manually labeled real images. Whereas the gap for cars was not significantly large, the gap for buses was considerable. The potential reason for the gap for buses was also due to an insufficient amount of training images that included buses.

To bridge the gap, the model was fine-tuned with a small number of real images tagged with true labels (= only 300 pairs of images) after being pretrained on the synthesized images. The test results of the fine-tuned model recorded almost the same performance as the model trained on a large amount of manually labeled images. More concretely, the observed and the estimated numbers of stopped vehicles turned out to be statistically identical at a 0.05 significance level. Table 3 shows that the p-values to

TABLE 1. Performance comparison in measuring average delays between the proposed CNN and reference models.

(a) Performance comparison for testbed 1

Period	0 ~ 15 min.		15 ~ 30 min.		30 ~ 45 min.	
Lane group	Straight-ahead	Left-turn	Straight-ahead	Left-turn	Straight-ahead	Left-turn
Pseudo-ground truth (sec)	29.38	51.53	28.09	67.83	22.77	59.04
Delay from the proposed model (sec)	29.23	51.50	27.98	68.00	22.67	59.22
Delay from a YOLO-based model (sec)	30.98	52.31	28.87	67.5	23.15	60.21
Delay from the HCM formula (sec)	27.27	48.00	25.79	61.04	24.98	47.27
Traffic volume for 15-min (veh)	237	120	211	143	194	118
Green time ratio (g/C)	55/140	28/140	55/140	28/140	55/140	28/140

Period	45 ~ 60 min.		60 ~ 75 min.		75 ~ 90 min.	
Lane group	Straight-ahead	Left-turn	Straight-ahead	Lane group	Straight-ahead	Left-turn
Pseudo-ground truth (sec)	21.40	59.02	28.53	68.12	27.51	69.74
Delay from the proposed model (sec)	21.26	59.14	28.36	68.33	27.48	69.78
Delay from a YOLO-based model (sec)	22.29	60.23	30.9	68.81	28.72	72.86
Delay from the HCM formula (sec)	24.72	51.47	27.66	60.25	27.39	59.48
Traffic volume for 15-min (veh)	188	128	243	142	239	141
Green time ratio (g/C)	55/140	28/140	55/140	28/140	55/140	28/140

(b) Performance comparison for testbed 2

Period	0 ~ 15 min.		15 ~ 30 min.		30 ~ 45 min.	
Lane group	Straight-ahead	Left-turn	Straight-ahead	Left-turn	Straight-ahead	Left-turn
Pseudo-ground truth (sec)	29.65	44.13	29.68	43.79	34.79	46.69
Delay from the proposed model (sec)	29.58	44.02	29.75	43.67	34.55	46.15
Delay from a YOLO-based model (sec)	30.07	44.69	31.24	44.34	33.94	44.00
Delay from the HCM formula (sec)	34.14	48.94	33.64	48.39	34.05	50.63
Traffic volume for 15-min (veh)	110	36	92	34	107	41
Green time ratio (g/C)	45/160	24/160	45/160	24/160	45/160	24/160

Period	45 ~ 60 min.		60 ~ 75 min.		75 ~ 88 min.	
Lane group	Straight-ahead	Left-turn	Straight-ahead	Lane group	Straight-ahead	Left-turn
Pseudo-ground truth (sec)	23.57	34.35	18.64	36.32	29.49	16.78
Delay from the proposed model (sec)	23.64	34.61	18.75	36.15	29.75	16.56
Delay from a YOLO-based model (sec)	23.83	36.20	19.07	36.41	30.55	17.82
Delay from the HCM formula (sec)	33.66	50.63	33.72	47.05	33.53	47.05
Traffic volume for 15-min (veh)	93	41	95	28	88	28
Green time ratio (g/C)	45/160	24/160	45/160	24/160	45/160	24/160

TABLE 2. Applicability comparison between the proposed model and a YOLO-based model.

	Proposed model	YOLO-based model
Weight size (MB)	10.9MB	236MB
Average computing time to count stopped vehicles for a pair of images (sec)	$1.13*10^{-2}$	$5.56*10^{-2}$

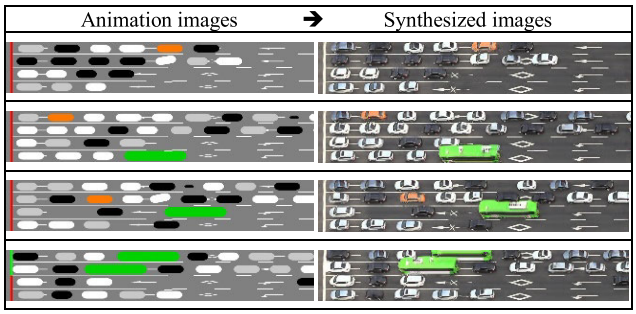
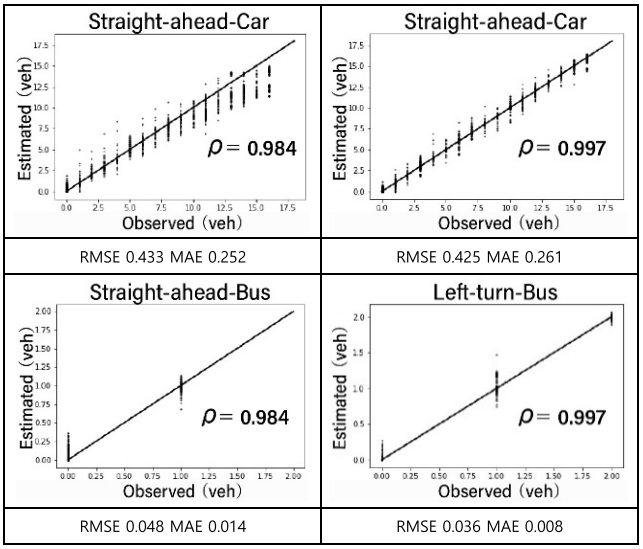


FIGURE 6. Conversion from naïve animation images to real-looking synthesized images.

reject the null hypothesis for two equal population means. When a value exceeded 0.05, the estimated result was not statically different from the observed number of stopped vehicles.

Fig. 8 shows a small discrepancy in delays between the pseudo ground truth and traffic delays, as measured by the fine-tuned model. These results verified that this scheme can minimize the human effort required to secure labeled data for training whenever the proposed CNN is applied to different sites.



(b) Test result for the fine-tuned mode after training on synthesized images

FIGURE 7. Performance of the model trained on synthesized images for counting the number of stopped vehicles (for testbed 1).

TABLE 3. Measuring the number of stopped vehicles using the model trained on manually labeled images compared with using the model trained on synthesized images.

		The model trained on manually labeled images		The model pretrained on synthesized images		The fine-tuned model after pretraining on synthesized images	
		Straight-ahead	Left-turn	Straight-ahead	Left-turn	Straight-ahead	Left-turn
C	RMSE	0.177	0.165	1.543	1.195	0.433	0.425
a	MAE	0.085	0.086	1.017	0.912	0.252	0.261
r	P-value	0.912	0.919	0.000	0.025	0.949	0.942
B	RMSE	0.011	0.033	0.250	0.430	0.048	0.036
u	MAE	0.002	0.009	0.111	0.207	0.014	0.008
s	P-value	0.777	0.637	0.000	0.003	0.269	0.833

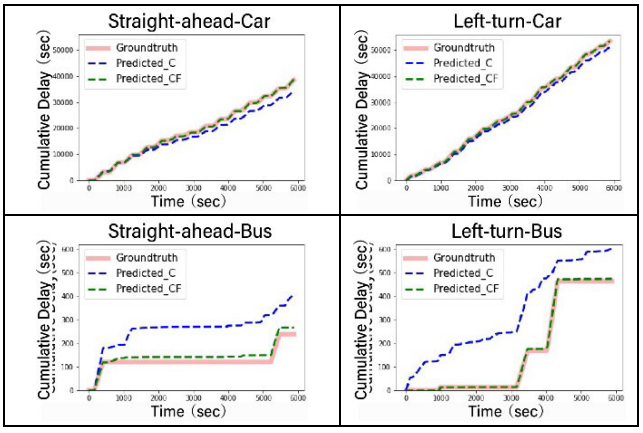


FIGURE 8. Performance comparison in measuring the traffic delays for testbed 1. "Predicted_C " represents delays estimated via the model pretrained on synthesized images, "Predicted_CF " represents those estimated from fine-tuned model after pretraining on the synthesized images.

VIII. CONCLUSIONS AND FURTHER STUDIES

The present study proved that traffic delays can be measured based solely on consecutive video images in an end-to-end manner. Once the number of stopped vehicles is counted using a deep CNN, measuring the delay is a trivial task of accumulating the count numbers over time.

The proposed model for measuring traffic delays outperformed the existing HCM formula and showed a comparable performance with a state-of-the-art vehicle detection algorithm. Furthermore, a plausible way to secure labeled images based on a traffic simulation and a CycleGAN was suggested. Owing to this scheme, the transferability of the proposed approach is guaranteed. Training the model takes little human effort to tag images with labels when it is applied to other road segments.

The proposed model also resolved the biggest problem in applying a RL algorithm for traffic-light control. Although delay has been a basic parameter that was used to determine the state and reward in a RL framework, no previous study has presented a robust method to measure it on a real-time basis. The proposed approach provides a plausible solution to set up a delay-based reward and penalty. That is, a RL algorithm is rewarded if the cumulative delay measured in the field decreases after an action is taken, whereas it is penalized if the delay increases. In our on-going study, the proposed approach to measure traffic delay is being fully incorporated with a RL algorithm for traffic light control in an urban arterial.

The proposed model was validated only for two different sites due to the difficulty of obtaining ground truth delays. If any third-party agency like a navigation company provided mobile information of probe vehicles, the delay estimated from the proposed methodology could be validated on a large scale.

REFERENCES

[1] Transportation Research Board, "Highway capacity manual," Nat. Res. Council, Washington, DC, USA, TRB Special Rep. 193, 2000.

[2] B. Abdulhai and L. Kattan, "Reinforcement learning: Introduction to theory and potential for transport applications," *Can. J. Civil Eng.*, vol. 30, no. 6, pp. 981–991, 2003.

[3] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[4] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Washington, DC, USA, Oct. 2011, pp. 1580–1585.

[5] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Stateline, NV, USA, vol. 25, Dec. 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[9] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1520–1528.

[10] P. Xu, Q. Yin, Y. Huang, Y. Z. Song, Z. Ma, L. Wang, T. Xiang, W. B. Kleijn, and J. Guo, "Cross-modal subspace learning for fine-grained sketch-based image retrieval," *Neurocomputing*, vol. 278, pp. 75–86, Feb. 2018.
[11] W. R. Hereth, A. Zundel, and M. Saito, "Automated estimation of average stopped delay at signalized intersections using digitized still-image analysis of actual traffic flow," *J. Comput. Civil Eng.*, vol. 20, no. 2, pp. 132–140, 2006.
[12] J. Chung and K. Sohn, "Image-based learning to measure traffic density using a deep convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1670–1675, May 2018.
[13] J. Lee, S. Roh, J. Shin, and K. Sohn, "Image-based learning to measure the space mean speed on a stretch of road without the need to tag images with labels," *Sensors*, vol. 19, no. 5, p. 1227, 2019.
[14] H. D. Robertson, *Manual of Transportation Engineering Studies*. Englewood Cliff, NJ, USA: Prentice-Hall, 1994.
[15] C. S. Asha and A. V. Narasimhadhan, "Vehicle counting for traffic management system using YOLO and correlation filter," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Mar. 2018, pp. 1–6.
[16] G. Lopez. *Traffic counter with YOLO and SORT*. [Online], Available: https://github.com/guillelopez/python-traffic-counter-with-yolo-and-sort
[17] C. Cheng, Y. Du, L. Sun, and Y. Ji, "Review on theoretical delay estimation model for signalized intersections," *Transp. Rev.*, vol. 36, no. 4, pp. 479–499, 2015.
[18] F. Qiao, P. Yi, H. Yang, and S. Devarakonda, "Fuzzy logic based intersection delay estimation," *Math. Comput. Model.*, vol. 36, nos. 11–13, pp. 1425–1434, 2002.
[19] Y. S. Murat and O. Baskan, "Modeling vehicle delays at signalized junctions: Artificial neural networks approach," *J. Sci. Ind. Res.*, vol. 65, no. 7, pp. 558–564, 2006.
[20] G. Deshmukh and N. J. Uke, "A systematic review of image based moving vehicle detection techniques," *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 45–47, 2016.
[21] R. P. Singh, P. Sharma, and J. Madarkar, "Compute-extensive background subtraction for efficient ghost suppression," *IEEE Access*, vol. 7, pp. 130180–130196, 2019.
[22] S. Buttan and K. Venugopal, "On-road moving vehicle detection by spatio-temporal video analysis of static and dynamic backgrounds," in *Ambient Communications and Computer Systems*. Singapore: Springer, 2018, pp. 703–715.
[23] P. Gajbhiye, N. Cheggoju, and V. R. Satpute, "Moving object detection and tracking in traffic surveillance video sequences," in *Recent Findings in Intelligent Computing Techniques*. Singapore: Springer, 2018, pp. 117–128.
[24] J.-F. Song, "Vehicle detection using spatial relationship GMM for complex urban surveillance in daytime and nighttime," *Int. J. Parallel Program.*, vol. 46, no. 5, pp. 859–872, 2018.
[25] K. Zhong, Z. Zhang, and Z. Zhao, "Vehicle detection and tracking based on GMM and enhanced camshift algorithm," *J. Elect. Electron. Eng.*, vol. 6, no. 2, pp. 40–45, 2018.
[26] Z. Dai, H. Song, X. Wang, Y. Fang, X. Yun, Z. Zhang, and H. Li, "Video-based vehicle counting framework," *IEEE Access*, vol. 7, pp. 64460–64470, 2019.
[27] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018.
[28] S. Seong, J. Song, D. Yoon, J. Kim, and J. Choi, "Determination of vehicle trajectory through optimization of vehicle bounding boxes using a convolutional neural network," *Sensors*, vol. 19, no. 19, p. 4263, 2019.
[29] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017 pp. 3645–3649.
[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.
[31] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian optimization of machine learning hyperparameters on large datasets," 2016, *arXiv:1605.07079*. [Online]. Available: https://arxiv.org/abs/1605.07079
[32] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, "Scalable Bayesian optimization using deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2171–2180.
[33] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: https://arxiv.org/abs/1607.08022
[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[35] G. Marcus, "Deep learning: A critical appraisal," 2018, *arXiv:1801.00631*. [Online]. Available: https://arxiv.org/abs/1801.00631
[36] Transportation Research Board, "Highway capacity manual," Nat. Res. Council, Washington, DC, USA, TRB Special Rep. 209, 1994.

JOHYUN SHIN is currently pursuing the degree with the Department of Urban Engineering, Chung-Ang University. His research interests include machine learning and visual recognition.

SEUNGBIN ROH is currently pursuing the degree with the Department of Urban Engineering, Chung-Ang University. His research interests include machine learning and visual recognition.

KEEMIN SOHN is currently a Professor with the Department of Urban Engineering, Chung-Ang University. His research interest includes the applications of artificial intelligence to transportation engineering and planning.