Doctoral Dissertation

Traffic Measurement and Control Methods using Autonomous Learning

Arnan Maipradit

Program of Information Science and Technology
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Professor Keiichi Yasumoto
Ubiquitous Computing Systems Lab.
(Division of Information Science)

Submitted on July 14, 2022

A Doctoral Dissertation
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of Engineering

Arnan Maipradit

Thesis Committee:
Professor Keiichi Yasumoto (Supervisor)
Professor Kenichi Matsumoto (Co-supervisor)
Associate Professor Hirohiko Suwa (Co-supervisor)

Traffic Measurement and Control Methods using Autonomous Learning*

Arnan Maipradit

Abstract

Traffic management is an essential part of intelligent transportation systems (ITS). It monitors and controls traffic to reduce congestion and to improve traffic flow. In order to implement appropriate traffic control, it is necessary to understand the traffic volume at more locations in real time, and to control the signal for efficient traffic scheduling. To improve traffic control, traffic control based on accurate traffic information and global traffic information is necessary. Therefore, in this thesis, we address these two challenges. To address the first challenge, we propose PAVEMENT, a novel autonomous incremental learning based traffic-census sensor system using a piezoelectric vibration sensor and a video camera without human intervention. PAVEMENT consists of two models: the video-based model which detects vehicles by using bounding boxes (detected by YOLOv3 and DeepSORT) and the vibration-based model which uses road vibrations to detect passing vehicles. To reduce the burden of collecting ground truth labels, we apply supervised learning to train the vibration-based model by using the result of the video-based model as ground truth. Once the vibration-based model is trained, it can be used for traffic census on roads without the video camera for various conditions. We collected the video and vibration data of more than 4,000 passing vehicles on roads in different places and applied our method to the data. As a result, PAVEMENT achieved over 98.4% accuracy and 98.0% f1-score in detecting passing vehicles with the model trained by 15 steps of incremental learning in 1-minute interval. To address the second challenge,

*Doctoral Dissertation, Graduate School of Science and Technology Science,
Nara Institute of Science and Technology, .

we propose an adaptive traffic control algorithm based on back-pressure and Q-learning. Adaptive traffic control is a strategy to control traffic signals based on actual traffic information. To improve adaptive traffic control, our approach uses back-pressure routing which was originally developed for routing packets based on queue length differentials (also called pressure gradients) in wireless communication networks. We apply back pressure routing to signal control at junctions regarding vehicles as packets. We also use Q-Learning to predict real-time traffic information at junctions and global traffic information that are input to our algorithm. We evaluated the proposed algorithm through computer simulations and confirmed that our algorithm reduces average vehicle traveling time from 17% to 38% compared with a state-of-the-art algorithm in some test scenarios.

Keywords:

autonomous incremental learning, passing vehicle detection, vibration sensor, LDA, traffic control, traffic measurement, back-pressure

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The increasing of urban vehicle cause traffic congestion. Implementation of appropriate traffic control will distribute congestion of the road network. In order to implement appropriate traffic control, it is necessary to understand the traffic volume on each road in real-time, and to control the signal for efficient traffic scheduling.

Currently, vehicle detection systems are widely used in ITS (Intelligence Transport Systems) due to their wide applicability to traffic management such as traffic monitoring and analysis traffic scheduling. Nowadays, many detection techniques use cameras and sensors to collect traffic data, such as vehicles' number, speed, and type. These techniques must produce results accurate enough to produce precise data for traffic management.

In implementing video-based passing vehicle detection, it is known that detection accuracy depends on weather conditions, shadows, camera blurriness, and obstacles [1–3]. Previous studies [4, 5] have used YOLO for object detection and Simple Online and Real-time Tracking (SORT) for tracking objects. However, the efficiency of the SORT algorithm decreases due to the occlusions. The DeepSORT algorithm introduced distance metric to improve the SORT algorithm [6]. Non-video-based passing vehicle detection methods using vibration sensors, Doppler sensors, and others are unaffected by weather conditions or time. However, getting ground truth data requires manual labeling, which is costly and labor-intensive.

Traffic congestion and increase in vehicle travel time occur because most traffic light control systems use fixed time cycle scheduling [7]. This is led by the algorithms in the control systems not considering real-time or real-situation traffic information. Congestion can be reduced by smartly controlling traffic signals [8]. With the development of technology of ITS and Internet of Things (IoT) [9], many researchers have adopted such technology to improve the efficiency of transportation. ITS is a traffic management system that uses an intelligent algorithm to reduce vehicle travel time and improve traffic safety.

As an implementation of the intelligent traffic control systems, SCOOT [10] and SCATS [11, 12] were studied. However, these adaptive traffic signals still cannot guarantee specific performance. In addition, decentralized algorithms are required to realize intelligent traffic control in a practical large-scale urban road network. Currently, decentralized traffic control algorithms have been proposed based on back-pressure [13–17]. Moreover, some back-pressure based algorithms have also been proposed to coordinate different vehicles [18]. In a road network, however, vehicles need time to travel from one road to another road which depends on the vehicle speed and road length. Directly applying a back-pressure algorithm is not appropriate to control traffic as in [18].

1.2 Related Works and Limitations

There are various methods for vehicle detection. Detection systems using computer vision detect vehicles by automatically analyzing real-time or recorded video from a camera [19–26]. Other detection systems use piezoelectric traffic sensors installed on the side of the road and detect vehicles by the voltage signal generated when a vehicle passes over the sensor [27–30]. When a vehicle's tires pass over the tube, pneumatic road tube sensors send a detection signal by air pressure along a rubber tube [31]. The induction loop method, in which a square wire is embedded under the road, detects the vehicle using the principle of magnetic fields introduced near electrical conductors causing electrical currents to be induced [32]. The approach using a magnetic sensor detects the vehicle that measures parameters of the magnetic field in that area [33–36]. Passive infrared devices detect vehicles by measuring the infrared energy radiating from the detection zone [30, 37]. Doppler microwave method or radio detection detects the vehicle based on reflected microwave or radio waves [38]. Acoustic detection systems provide traffic information based on noise generated by passing vehicles through a network of recorders [39, 40].

Several methods have high accuracy, but each has its drawbacks. Video Image Processing is affected by inclement weather, shadows, and poor lighting. The infrared and ultrasonic methods affect environmental conditions such as rain, fog, temperature, humidity, and air turbulence. The inductive loop method and the magnetometer method require a pavement cut and a lane closure to install. Microwave Radar requires a license for operation and maintenance. Acoustic detection systems can not distinguish between vehicles on the adjacent lanes when they generate noise at various levels of intensity.

In summary, existing methods have a drawback such as installation cost, installation process, maintenance process, and environmental condition, while some have advantages such as flexible re-location, cost-effectiveness, accuracy, and robustness to the environment. Primarily, computer vision-based methods are cost-effective as they can count in many directions at once but have the drawback that weather conditions impact their accuracy. On the other hand, non-vision-based methods are robust against weather conditions but suffer from ground truth data for training. In our approach, we propose a new method that combines the advantages of the computer vision-based method and piezoelectric sensor-based method to achieve robustness against weather conditions and easy deployment.

The intelligent traffic control systems have been implemented in an urban road network such as SCOOT [10] and SCATS [11, 12]. These systems use adaptive traffic signals which consider real-time traffic information [41] to become more effective than a fixed cycle signal control. However, these adaptive traffic signals still cannot guarantee global optimality. Genetic Algorithm [42] and Fuzzy Logic Control [43, 44] are also considered as the solution to smartly controlling traffic signals. However, these algorithms are centralized and do not suit a large urban road network that has many entities and requires decentralized algorithms.

Dynamic vehicle routing problems have also been widely studied [45]. The earlier literature only allows vehicles with some minor adjustments to the prior routes [46, 47]. With the development of technology, researchers started using Markov Decision Process to route vehicles dynamically without any prior route [48, 49]. Unfortunately, this method failed to be applied in the relatively large-scale road networks which exist mostly in the real world. To tackle this limitation, an approach based on Approximate Dynamic Programming has been proposed [50], yet all of the solutions above do not integrate with adaptive signal control. Recent research considered adaptive signal control and dynamic vehicle routing [51,52]. However, they only focused on providing adaptive route guidance for individual vehicles, not coordinating different vehicles. With the development of self-driving technology, it will be more efficient to coordinate different vehicles to reduce overall traffic congestion.

Recently, decentralized traffic control algorithms have been proposed based on back-pressure [13–17], as the back-pressure based traffic signal control algorithm shows superior performances to the signal control of fixed time cycles. These back-pressure based traffic control algorithms do not consider the adaptive control of vehicle routes, e.g., the shortest path algorithm easily results in traffic congestion especially during rush hours. Some research consider jointly controlling traffic signals and vehicle routing [14, 52]. However, these works only focus on giving individual vehicles adaptive route guidance. Coordination between different vehicles will further reduce traffic congestion.

Some back-pressure based algorithms have also been proposed to coordinate different vehicles [18]. In a road network, however, vehicles need time to travel from one road to another road which depends on the vehicle speed and road length. Directly applying a back-pressure algorithm is not appropriate to control traffic as in [18].

Compared to the existing back-pressure based algorithms mentioned above, this thesis is positioned to an adaptive traffic control algorithm which uses a back-pressure algorithm by considering vehicle traveling time on a road. Specifically, our algorithms control traffic signal and vehicle routes based on real-time traffic information such as the vehicle speed and vehicle position. As a result, our algorithm significantly reduces traffic congestion. In addition, not only based on local traffic information, i.e., every control agent considers information of vehicles around its own junction, this thesis covers more efficient traffic control which uses global traffic information and coordination between different junctions.

1.3 Thesis Statement and Contributions

In this dissertation, we introduce the methods that can distribute traffic congestion in road networks using autonomous learning. In order to implement appropriate traffic control, it is necessary to understand the traffic volume and control the traffic signal for efficient traffic scheduling. Therefore, in this thesis, we address these two challenges. To address the first challenge, we propose PAVEMENT, a novel autonomous-learning traffic-census sensor system using a piezoelectric vibration sensor and a video camera. To address the second challenge, we propose an adaptive traffic control algorithm based on back-pressure and Q-learning. Adaptive traffic control is a strategy to control traffic signals based on actual traffic information.

1.4 Organization of the Thesis

This dissertation shows how traffic measurement and control methods using Autonomous Learning can distribute and reduce traffic congestion in a road network. The organization of this thesis is as follows.

Chapter 2 introduces our PAVEMENT: self-learning passing vehicle detection system with camera and vibration-based Reinforcement Learning in detail. We first describe the architectural model, as well as provide our approaches. Then we show an experimental evaluation and the result.

Chapter 3 introduces our adaptive traffic control algorithm based on back-pressure and Q-Learning. We first describe the road network system, as well as provide our back-pressure and Q-learning algorithm. Then we show an experimental evaluation and the result. In Chapter 4, we present our conclusions.

2 PAVEMENT: Passing Vehicle Detection System with Autonomous Incremental Learning using Camera and Vibration Data

2.1 Introduction

This chapter proposes PAVEMENT (*passing vehicle detection by autonomous incremental learning*), a novel traffic-census sensor system using a piezoelectric vibration sensor and a video camera, to solve the weather condition and manual labeling problems in existing methods.

PAVEMENT consists of two models: the video-based model that detects vehicles using bounding boxes and the vibration-based model that uses road vibrations to detect passing vehicles. To reduce the burden of collecting ground truth labels, we use the result of the video-based model as ground truth to train the vibration-based model by applying linear discriminant analysis and incremental learning. Once the vibration-based model is trained on a road, it can continue detecting vehicles passing through the road without the video camera regardless of conditions like weather, lighting, and other environmental factors. For the video-based model, we have developed a vehicle detection model using YOLOv3 [53] which detects objects in each video frame. We also use DeepSORT [54] which is a real-time tracking algorithm for multiple 2D objects over subsequent video frames. Once the vehicle is detected and tracked over several frames, a simple mathematical calculation is applied to count the vehicles where an orthogonal line is defined on the road, and the intersection of the detected object (vehicle) and the line is calculated so that the number of vehicles passing the defined line is obtained. This is used as ground truth for training the vibration-based model. For the vibration-based model, we use data collected from a traffic census sensor (vibration sensor) and the output of the video-based model to train the model by applying linear discriminant analysis and incremental learning. Linear discriminant analysis (LDA) is implemented to reduce feature dimensionality of vibration data for later classification. The typical implementation of the LDA technique requires that all the samples are available in advance. However, there are situations where the entire data set is not available and the input data are observed as a stream. In this case, the LDA feature extraction should have the ability to update the computed LDA features by observing the new samples without running the algorithm on the whole data set. Thus, incremental learning is implemented to further train the model.

Evaluation experiments were conducted using a video camera and a vibration sensor on a road in Osaka city, Japan. The video camera collected day-time data while the vibration sensor collected day and night time data. The developed video-based model achieved 99% accuracy of passing vehicle detection. With the detection results of the video-based model as the ground truth, we applied the proposed method to incrementally train the vibration-based model. As a result, the trained vibration-based model showed a saturated accuracy and F1-score after applying 15 incremental learning steps in 1 minute interval, with a 98.4% accuracy and 98.0% F1-score in detecting passing vehicles.

2.2 Passing Vehicle Detection: Requirements and Approaches

Our goal is to realize a system for detecting (counting) passing vehicles on the road. The requirements are as follows.

R1: The system can detect passing vehicles accurately on a target lane regardless of time (day or night) and weather conditions (clear, rain, etc).

R2: The system is easy to deploy and maintain (installation and re-location are easy).

R3: The system does not require manual labeling for ground truth data collection.

R4: The system can adapt to any different road by autonomously learning from the data collected on the road without human intervention.

To fulfill the requirements R1–R4, we take the following approaches.

- For R1 and R2, we utilize a Piezoelectric vibration sensor module developed in [28] that can be easily deployed at the road side and capture the road vibrations generated when vehicles pass near the module on the road.
- For R3, we develop a video-based passing vehicle detection model to automatically get ground truth labels using image analysis and object detection.
- For R4, on each new road where a vibration sensor module and a video camera are deployed, we train a passing vehicle detection model based on road vibration data by applying LDA and incremental learning where labels (ground truth) are obtained by the video-based model.

2.3 PAVEMENT: Self-Learning Passing Vehicle Detection System

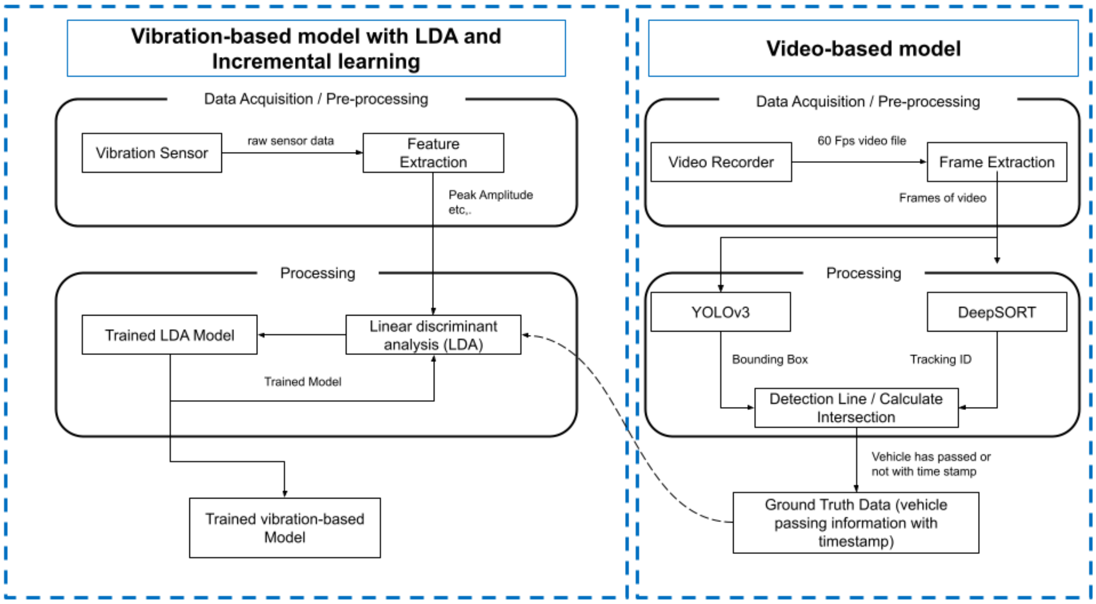

Figure 2.1: PAVEMENT: System Architecture.

In this section, we describe our approach PAVEMENT for vehicle detection with a vibration sensor and video camera based on LDA and autonomous incremental learning.

2.3.1 Outline

The PAVEMENT system automatically collects the ground truth labels (i.e., passing vehicles) from video camera data using image analysis [22]. We collect data for detecting passing vehicles by using a vibration sensor in the same way as a traffic census sensor [28]. PAVEMENT uses Linear discriminant analysis (LDA) and Incremental Learning to train the vibration-based model. LDA is a generalization of Fisher's linear discriminant, a method used in statistics and other fields, to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before classification. Incremental Learning is a method of machine learning in which input data is continuously used to extend the existing model's knowledge i.e. to further train the model. It represents a dynamic technique of supervised learning and unsupervised learning that can be applied when training data becomes available over time. With the proposed system, the passing vehicle detection model using vibration data is autonomously trained without human intervention by setting the camera and the vibration sensor on the roadside. Once the vibration-based model is trained, it is used to detect passing vehicles without cameras.

The architecture of the proposed system is shown in Fig. 2.1. PAVEMENT system consists of two parts: 1) video-based passing vehicle detection model (right of Fig. 2.1), and 2) Vibration-based model (left of Fig. 2.1). The details of these parts are described in the following sections.

2.3.2 Video-based Passing Vehicle Detection

As shown in Fig. 2.1 (right), the proposed video-based passing vehicle detection method utilizes YOLOv3 [53] for object detection and DeepSORT [54] for object tracking. The detection of vehicle passing is performed by checking if the bounding box of the vehicle crosses the detection line drawn on the road in video frames.

The goal of object tracking is to locate the position of a moving object in one video frame. Object detection is the first phase in any object tracking program. The purpose of object detection is to determine whether any instance of an object is in an image or video frame. After object detection, the tracking operation is used to find the positions of the detected objects in each frame. The input is the bounding box of an object detected by YOLO. The tracking phase will track the detected objects until the end of the video stream.

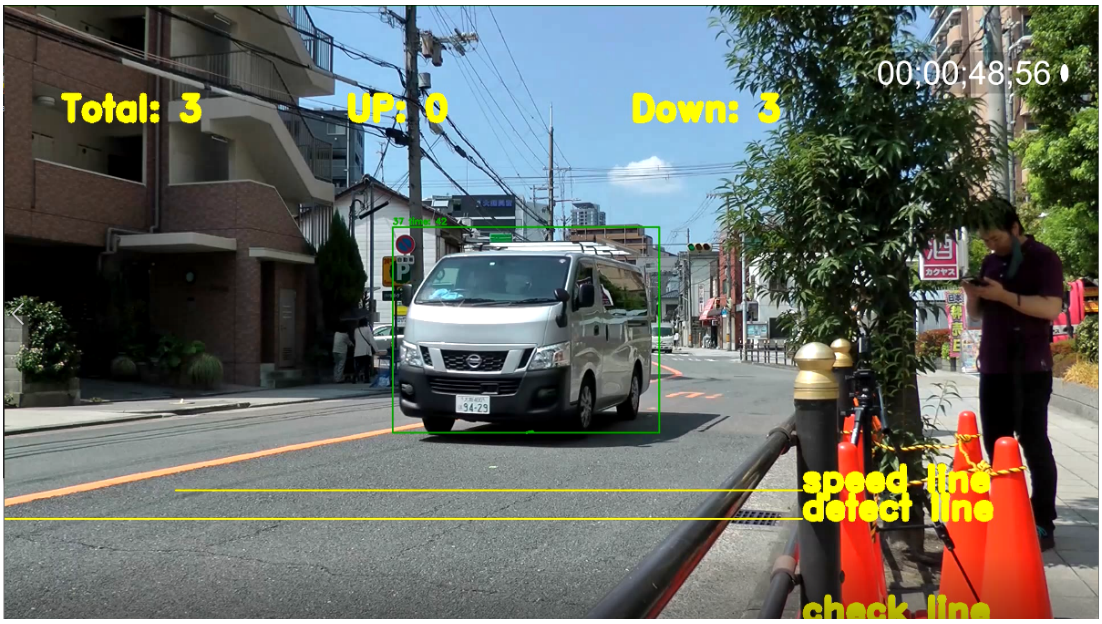

Figure 2.2: Video-based Passing Vehicle Detection.

An example of our video-based passing vehicle detection is shown in Fig. 2.2, where a *Total* shows the number of vehicles (sum of upstream and down-stream) that passed *detect line*, and *Up* and *Down* show the number of vehicles on the upstream (opposite direction) lane and downstream (forward direction) lane, respectively. The developed method also collects vehicle speed by using timestamp and distance from *speed line* and *check line* for each vehicle. When a vehicle passes through *detect line*, the system records a timestamp.

The procedure of the developed method is as follows:

1. import video file from a video camera, loop over frames from the video file stream, construct a blob from the input frame and then perform a forward pass of the YOLO object detector that gives us bounding boxes of vehicles and associated probabilities.

2. initialize the list of detected bounding boxes, confidences, and class IDs, respectively, and extracts the class ID and confidence (i.e., probability) of the object detection.

3. filters out weak predictions by ensuring that the detected probability is

greater than the minimum probability.

4. scale bounding box coordinates relatively to the size of the image and updates the list of bounding box coordinates, confidences, and class IDs.

5. applies non-maxima suppression to suppress weak, overlapping bounding boxes.

6. extracts the bounding box coordinates, draw a bounding box rectangle, and labels on the image.

7. draws detection line and checks if a bounding box intersects with the detection line with linear system calculation.

2.3.3 Vibration-based model with Incremental Learning

As shown in Fig. 2.1 (left), the proposed method for training vibration-based model utilizes a vibration sensor, LDA, and incremental learning by using the vibration data and the ground truth labels obtained by the video-based model.

Vibration Sensor

We use a piezoelectric vibration sensor system [28] which is developed to collect traffic census data. The collected data can be used to count passing vehicles on the road. This system is based on a piezoelectric vibration sensor that senses road vibrations from passing vehicles by deploying the system on sidewalks next to the target road. The system consists of a piezoelectric sensor unit, an amplifier, and an audio interface. As a sensor unit, a 7BB-41-2L0 sensor (Murata Manufacturing) is used to convert vibrations into electrical voltage signals in this system. The system employs the amplifier which has an impedance conversion circuit to improve the signal-to-noise ratio (SNR) of the input signal. Finally, UA-25EX Universal Serial Bus (USB) (Roland DG Corp.) is utilized as an audio interface to convert analog signals from the amplifier into pulse code modulation (PCM) signals (16 bit, 44.1 kHz). To avoid clipping the input signal, the gain of the USB is adjusted so that a –15 dBV sinusoidal 1 kHz analog signal is recorded as 0 decibels relative to the full scale (dBFS). Signal data was then recorded to a PC from the vibration sensor as audio data.

Data Preprocessing

We extract the vibration peaks from the acquired vibration data in a fixed interval (e.g., 1 minute) using the algorithm proposed in [28]. Only the extracted vibration peaks data is input to incremental learning. The process of detecting vibration peaks is as follows.

- Apply a short-time Fourier transform to vibration data with a window length of 16,384 samples (about 0.37 s) and a hop length of 8,192 samples (about 0.19 s).
- Exclude frequency components higher than 300 Hz from short-time Fourier transformed data.
- Mark and record timestamp of peak points in energy trend data that exclude from previous step.
- For each peak point, extract 2 second of vibration data so that the peak point is located at the center of the data.

Preprocessing

We extract 8 features from each 2 seconds data including the peak point. We apply Mel-frequency cepstral coefficients (MFCCs) since vehicle sound data is similar to a human voice signal in terms of frequency range. The process reshapes the matrix (8×173) to a vector (1×1384), PAVEMENT uses this data as an input to Linear discriminant analysis (LDA).

Linear discriminant analysis (LDA)

Listed below are the 5 general steps for performing LDA.

- Compute the d-dimensional mean vectors for the different classes from the data set.
- Compute the scatter matrices (in-between-class and within-class scatter matrix).
- Compute the eigenvectors ($e_1, e_2, ..., e_d$) and corresponding eigenvalues (λ_1, $\lambda_2, ..., \lambda_d$) for the scatter matrices.
- Sort the eigenvectors in decreasing order and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix W (where every column represents an eigenvector).
- Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $Y = X \times W$ (where X is a $n \times d$-dimensional matrix representing the n samples, and y are the transformed $n \times k$-dimensional samples in the new subspace).

To apply LDA, we assign one of the two data label classes below to each extracted 2 second data including the peak point.

- 1 (a vehicle is passing through the target measurement lane)
- 0 (a vehicle is not pass through the target measurement lane; a vehicle may be passing through the other, e.g., opposite lane)

Here, labels are given automatically by the video-based passing vehicle detection system.

This LDA process generates a new axis onto which it projects data in a way that minimizes the variance and maximizes the distance between the means of the classes. Since PAVEMENT inputs a time series data, we apply incremental learning to LDA. In our approach, we use labels (given automatically), features extracted from a fixed time interval (e.g., 1 minute) of vibration data and a model already trained at the previous intervals data to update the model until the classification accuracy is saturated.

2.4 Experimental Evaluation

This section shows the experimental environment and the dataset used for evaluation and then the performance of the video-based detection model and the vibration-based detection model trained by LDA and Incremental learning. Our approach uses a road vibration sensor [28] developed as non-intrusive detector which can be mounted on a structure above the surface of the pavement. We use performance index consisting of Recall, Precision, F1-score, and Accuracy.

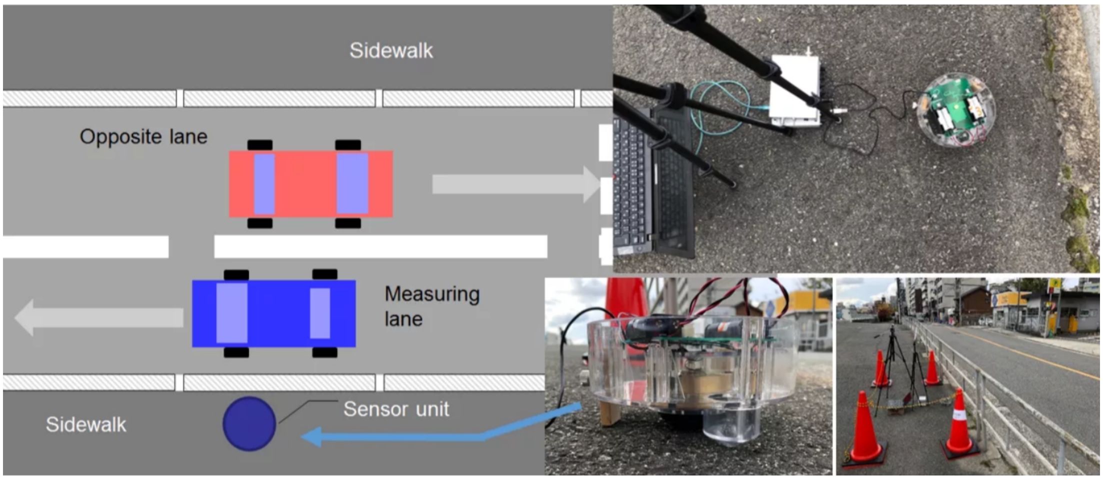

Figure 2.3: Piezo sensor deployment location.

2.4.1 Experimental environment and dataset

We collected video data and vibration data on an actual two-lane road in Osaka city, where a video camera and a vibration sensor system are deployed as shown in Fig. 2.3. Fig. 2.4 and Fig. 2.5 show the photos of the road with a video camera and a vibration sensor system deployed.

Figure 2.4: System deployment during training phase.

Figure 2.5: System deployment during operation phase.

We collected the following video and vibration data.

- Video: 1920×1080 pixels and 60 frames per second, Duration: daylight hours, Locations: three

- Vibration: raw PCM data with 16bit, 44.1kHz, Duration: day and night time, Locations: two

We extracted the common time period where both the video data and the vibration data are available and created the dataset of the common time period for autonomous incremental learning. The created dataset includes the data of 7,147 passing vehicles. The dataset was divided into two parts with a ratio of 7:3 for training and test.

2.4.2 Performance of video-based passing vehicle detection

Fig. 2.6 shows the procedure to count the passing vehicles from the recorded video.

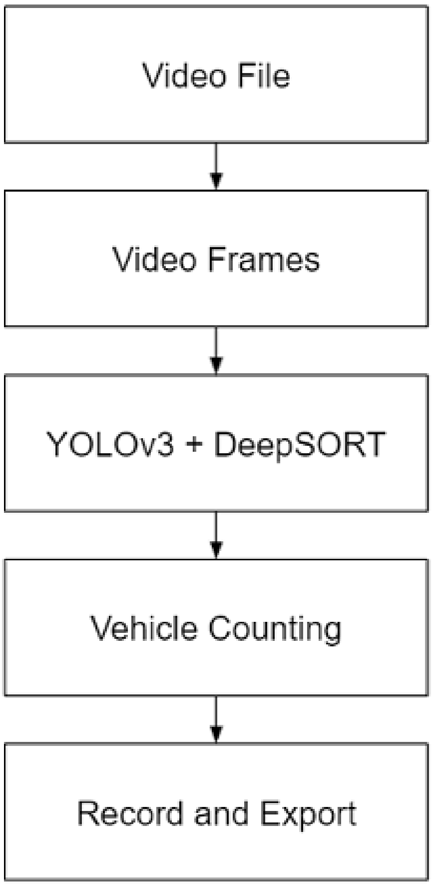

Figure 2.6: Video-based Passing Vehicle Detection process.

Computation time

Experiments were run on a computer with an Intel Core i5–8400 2.8 GHz with 6 cores and 6 threads, GPU NVIDIA GTX 1080, RAM 32GB, Windows 10 OS. Video-based passing vehicle detection took up to 60 minutes to process each 15-minute video. It means that we cannot train the model in real time. However, our proposed approach is still feasible by intermittently training the model, for example, using 15 minutes video of each hour for training. If we use faster PC, we can shorten this intermittent period.

Results

We tested video-based passing vehicle detection with real video camera recordings from Osaka prefecture consisting of over 100 videos collected at three locations during daytime, each containing 15 minutes of footage and has more than 10,000 vehicles. We achieved 99% accuracy for passing vehicle detection and were also able to collect the speed of vehicles. The 1% missing was found to occur when a smaller vehicle stays too close to a bigger vehicle, making our system unable to detect that vehicle as shown in Fig. 2.7. To achieve higher accuracy video camera angle should be more on the side of the vehicle so it can record every vehicle passing through the video camera or filter a bicycle and motorcycle out of the measurement target. Table 2.1 shows the detail of detection results. There is no false positive and the false negative is also reasonably small (87 among 11,742). The F1-score is 99.63%.

Table 2.1: Results of Video-based Passing Vehicle Detection

Parameters	Values
`Actual counts`	11,742
`Detected counts`	11,655
`False Positive`	0
`False Negative`	87
`Precision`	100%
`Recall`	99.25%
`F1`	99.63%
`Accuracy`	99.26%

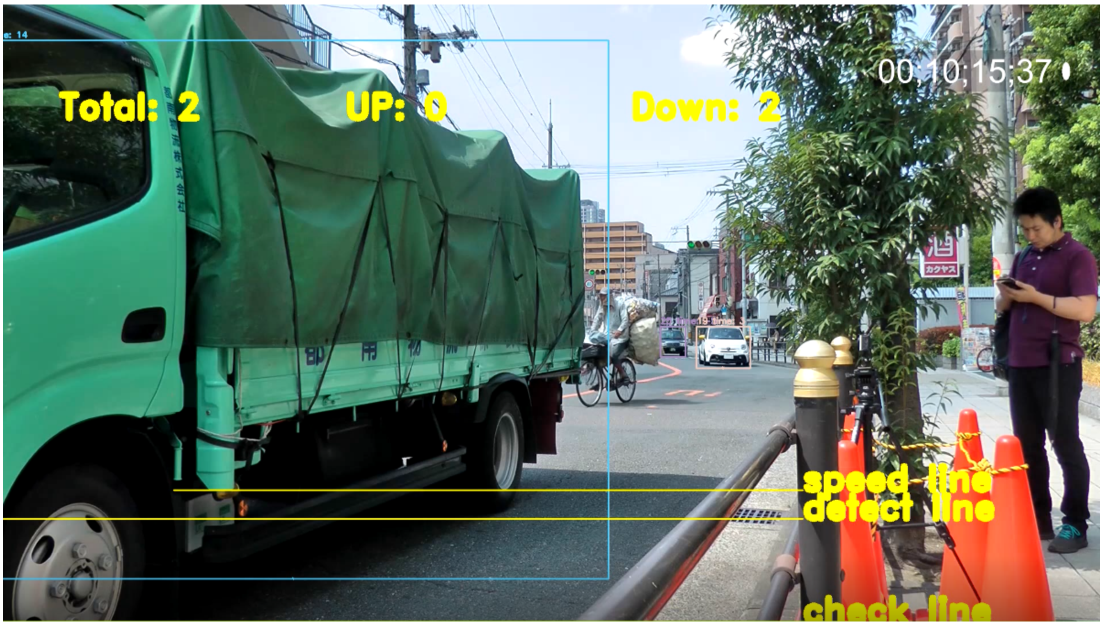

Figure 2.7: False negative situation in Video-based passing vehicle detection.

2.4.3 Vibration-based model with incremental learning

We applied the proposed method to use incremental learning to train vibration-based model to the dataset.

Fig. 2.4 shows deployment of our system during the training phase that consists of a video camera to collect ground truth data and a vibration sensor to collect vibration data. Once the vibration-based model reaches the saturated accuracy, a video camera can be removed as shown in Fig. 2.5.

In this experiment, we assume that the video-based vehicle detection gives labels in real time.

Computation time

Experiments were run on a computer with an Intel Core i5–8400 2.8 GHz with 6 cores and 6 threads, RAM 32GB, Window 10 20h1 os. Incremental LDA process at each 1 minute interval took up to 18.7 seconds to process 1 minutes of vibration data. It also took up to 4 seconds to test the trained model to know the accuracy, the f1-score, etc. Thus, it is possible to incrementally train the model in real-time.

Results

Table 2.2 and Fig. 2.8 show the detail of detection results. As seen, the accuracy saturated when it reaches 15 minutes point.

Results show that accuracy and F1-score gradually as incremental learning steps proceed as shown in Fig. 2.8. After 15 minutes point of vibration data, it saturated and got 98.4% accuracy and 98.0% F1-score which are comparable to the existing method using the same vibration sensor with manually labeled ground truth data [28] which achieved 98.3% accuracy of passing vehicle detection.

Table 2.2: Results of Vibration-based model Vehicle Detection

Total minutes	Accuracy	Precision	Recall	F1-score
1	0.873	0.900	0.900	0.891
2	0.880	0.875	0.875	0.866
3	0.911	0.909	0.952	0.923
4	0.918	1.000	0.960	0.912
5	0.882	0.900	0.900	0.900
6	0.986	1.000	0.966	0.982
7	0.987	1.000	0.970	0.985
8	0.989	0.979	0.974	0.987
9	0.989	1.000	0.976	0.988
10	0.991	1.000	0.979	0.989
11	0.985	1.000	0.980	0.988
12	0.986	0.982	0.964	0.985
13	0.984	0.982	0.965	0.986
14	0.985	0.983	0.967	0.983
15	0.984	0.985	0.976	0.980

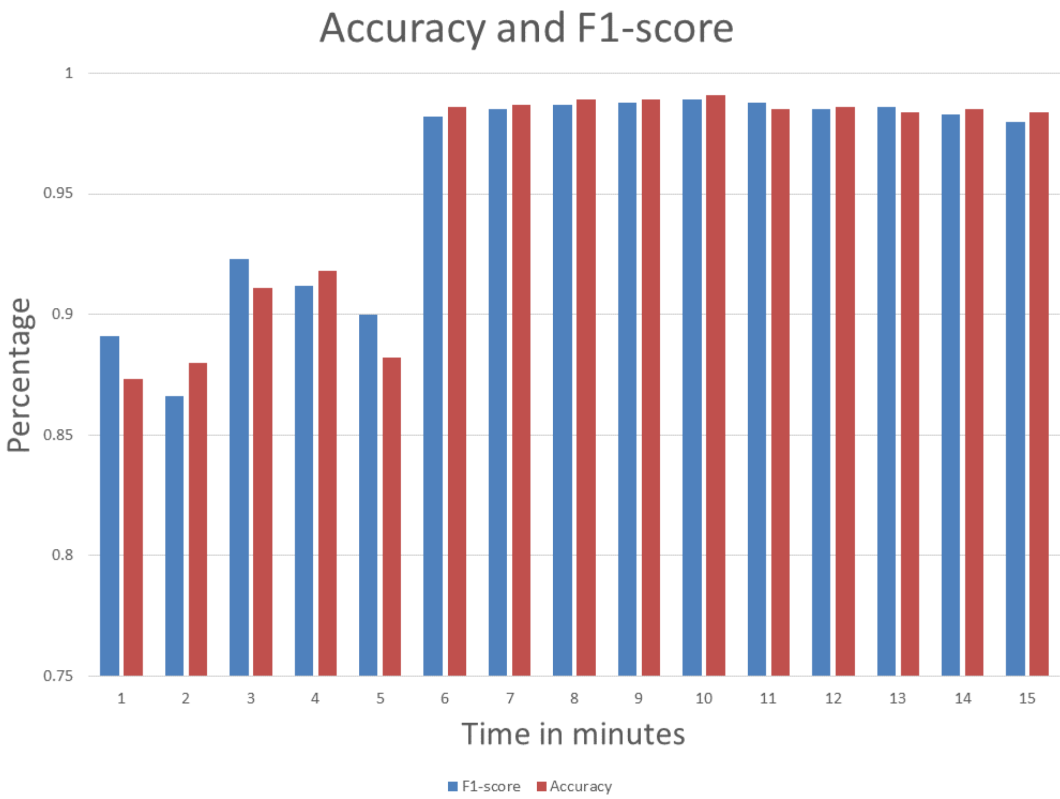

Figure 2.8: Accuracy and F1-score.

Our approach is also suitable for both day-time and night-time while the accuracy of video-based passing vehicle detection decreases in night-time situations. We use the same data with our algorithm as input in Image processing techniques, existing global detection method [55], Visual Saliency & Deep Learning [56], Segmentation Techniques [57], Weighted Feature Fusion and R-CNN [58], Feature extraction algorithm, pairing algorithm [59]. For automatic multilevel thresholding and matching algorithm [60] we can not implement these algorithms to test the accuracy so we use their experimental result to compare with other algorithms. Previous techniques have a detection rate in the range of 81-97%. In comparison, our approach's result shows that it can achieve 98% accuracy in day-time and a similar accuracy could be achieved for night-time situations since the vibration-based method will not be affected very much by weather and light conditions. As a result, our approach could achieve high accuracy and is suitable for all weather conditions.

3 Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning

3.1 Introduction

In this chapter, we propose an adaptive traffic control algorithm which uses a back-pressure algorithm by considering vehicle traveling time on a road. Specifically, our algorithm controls traffic signals and vehicle routes based on real-time traffic information such as vehicle speed and vehicle position. As a result, our algorithm significantly reduces traffic congestion. In addition, not only based on local traffic information, i.e., every control agent considers information of vehicles around its own junction, we also propose another adaptive traffic control algorithm which uses global traffic information and coordination between different junctions. The latter algorithm controls traffic based on accurate real-time traffic information and local traffic information to global traffic information, where neighboring junction agents exchange traffic information to learn global traffic information.

The proposed algorithms were previously presented in [61] and [62], respectively. This chapter consists of an explanation of the proposed algorithms and experiment results with additional results. Section 3.2 introduces a road network system assumed in this chapter. Section 3.3 explains our proposed methods, and experiment results are shown in Section 3.4.

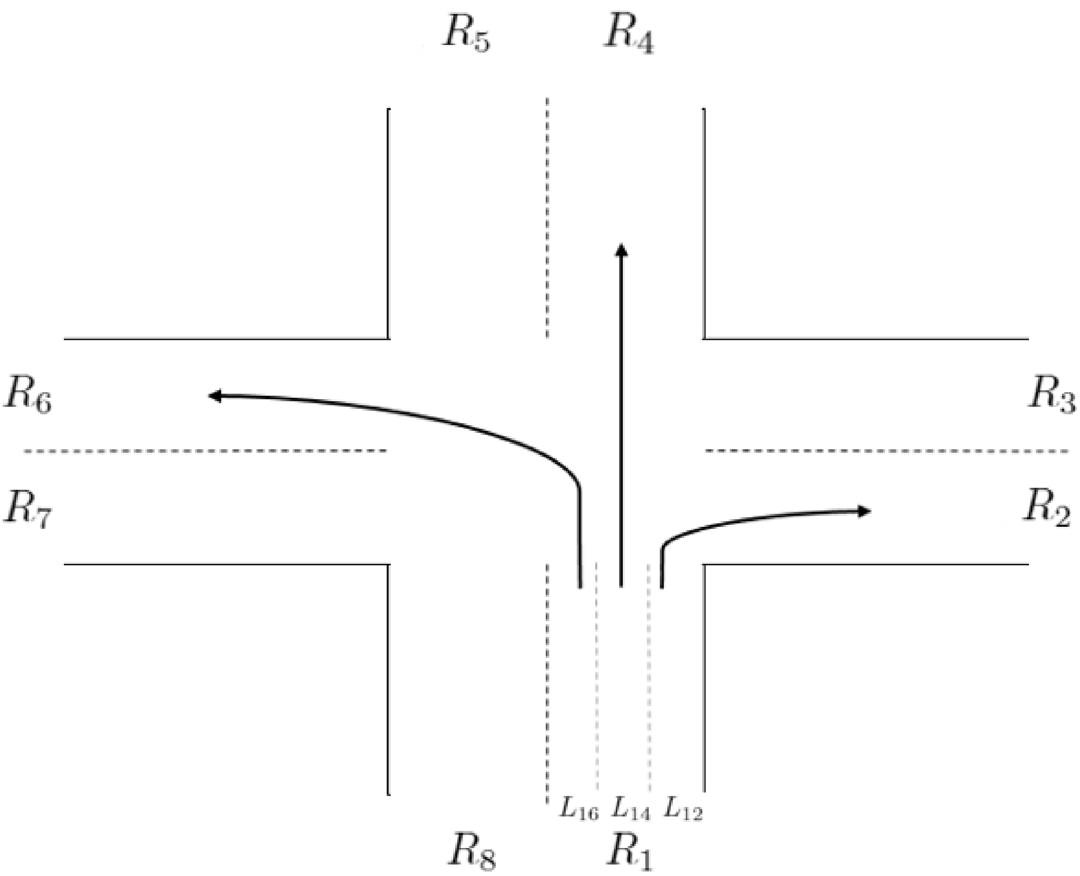

Figure 3.1: An example of a junction with roads of three lanes.

3.2 Road Network System

A road system consists of Roads (R) and Junctions (J), where $\mathbb{R} = \{R_1, R_2, R_3, ..., R_{max}\}$ denotes roads, $\mathbb{J} = \{J_1, J_2, J_3, ...J_{max}\}$ denotes junctions. It is assumed that each R_i consists of 3 lanes L_{ij}, an example is given in Fig. 1. Vehicles of a traffic flow (f) have the same starting road (o) and destination road (d). We define $\mathbb{F}$ as the set of all flows, $\mathbb{O} = \{o(f), f \in \mathbb{F}\}$ as the set of all starting roads, $\mathbb{D} = \{d(f), f \in \mathbb{F}\}$ as the set of all destinations and $\lambda_f(t)$ as the number of vehicles of flow f that enter road network at time slot t.

We define a traffic movement (R_i, R_j) at a junction to be the process of a vehicle moving from R_i to R_j. We define a traffic phase to include all traffic movements that can happen simultaneously. Fig. 2. shows all possible phases at

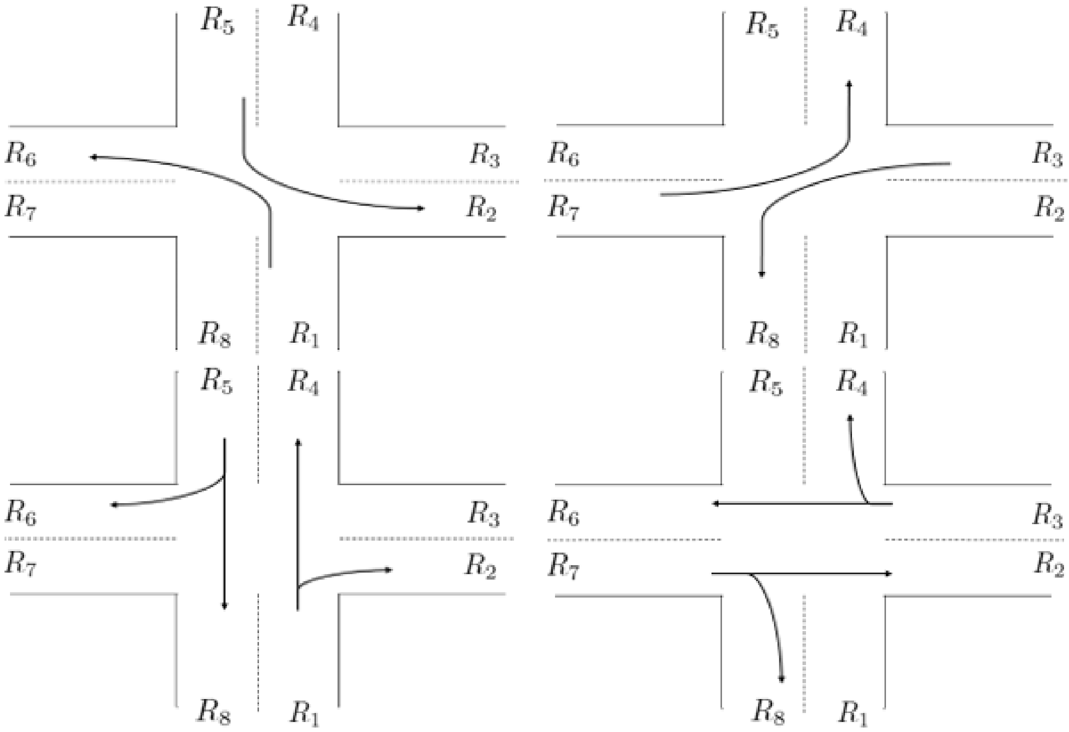

Figure 3.2: All possible phases at a junction.

a junction. For a junction J_a, we define $\mathbb{M}_a$ as the set of all possible movements and $\mathbb{P}_a$ as the set of all possible traffic phases. Traffic signals at junction J_a are controlled by activating a traffic phase p_i^a from $\mathbb{P}_a$.

3.3 Adaptive Traffic Signal Control Based on Back-Pressure with Global Information

In this section, we describe our proposed algorithms based on back-pressure with global information.

3.3.1 Overview

We have proposed adaptive traffic signal control methods based on back-pressure with global traffic information [61, 62]. [61] describes an adaptive traffic signal control method using real-time traffic information with global traffic information in a road network. We assume all vehicles are self-driving vehicles that are amounted with accurate speed sensors and GPS devices, and can communicate in a timely way with control agents via networks, like vehicle-to-vehicle (V2V), vehicle-to-Road Side Unit (V2R), etc. The control agents are the computer programs placed at each junction to collect information of vehicle speed and vehicle position at every time slot for traffic control. A vehicle is also able to provide traveling time to control agents. The traveling time includes not only moving time but also waiting time to turn in junctions. At each time slot, every control agent performs the following three tasks sequentially. In addition, the communication loads are greatly increased by control agents if each agent widely exchanges information. Therefore, [62] employs the approach in which control agents are placed on static equipment at each junction and exchange information only with their neighboring agents. Q-learning is used to estimate global congestion information from limited information. The procedures in our proposed method using the estimated information are as follows;

1. Task 1. Learning Global Congestion Information:
   Control agents exchange congestion level information with their neighboring agents to maintain a table of values $R^d_{ij}(t)$ because the table of values is usually different among neighboring agents. Congestion level information of a road is an index of congestion and is defined by the number of shadow vehicles at a shadow queue associated with the road. Based on the exchanged congestion information, the agents update their congestion estimates based on Q-learning. Through exchanging and updating congestion information, all agents finally obtain global congestion information from recursive definition (1) in Section 3.3. From recursive definition (1), we can see that $R^d_{ij}(t)$ involves all other $R^d_{ij}(t)$ values, thus called global congestion information. Although each road is assigned to two different agents at its junctions, this thesis assumes that global congestion information for a specific road obtained by the two assigned agents is not so different because used information is periodically synchronized among neighboring agents. Global congestion information is used in the following two tasks.

2. Task 2. Traffic Phase Selection:

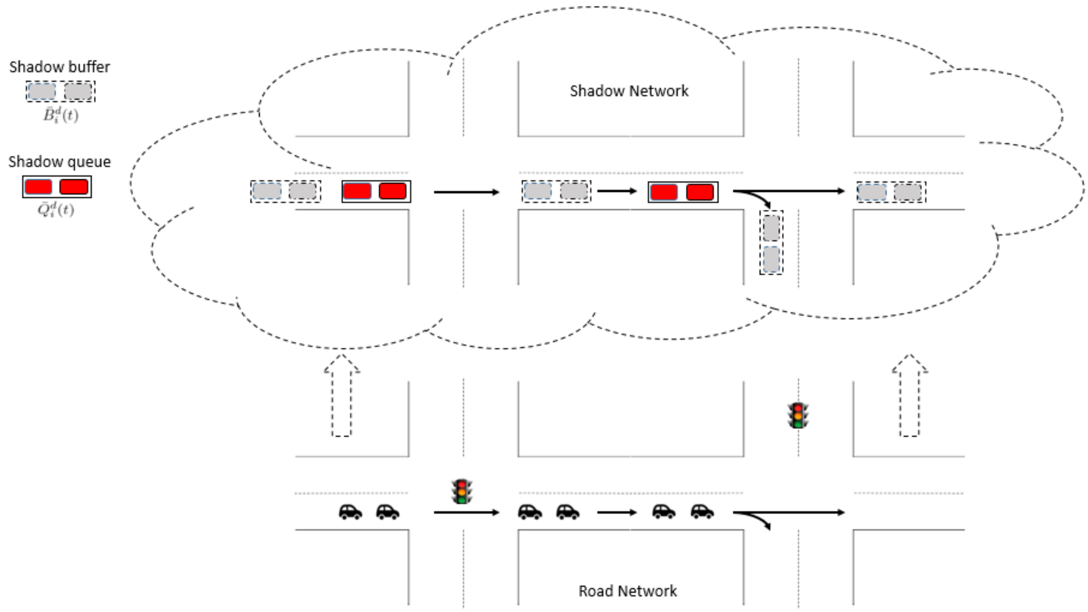

Figure 3.3: An example of a shadow network.

The agent selects a traffic phase based on the back-pressure algorithm.

3. Task 3. Vehicle Routing:
   After a vehicle passes through the junction and enters the next road under the traffic phase selected in task 2, the agent determines which lane of that road the vehicle should join. Since each lane determines the vehicle turning direction, i.e., going straight, turning left, or turning right, the process of determining lanes for a vehicle to join forms the routing process of that vehicle. The following shadow network is constructed to perform these three tasks.

3.3.2 Shadow Network

An example of a shadow network is given in Fig. 3, where a virtual shadow vehicle in a shadow network corresponds to an actual vehicle in a road network, a shadow buffer $\bar{B}_i^d(t)$ corresponds to the beginning part of one real road (a vehicle just passing through a junction will enter this part of the road) and a shadow queue $\bar{Q}_i^d(t)$ corresponds to the end part of one real road (a vehicle running close

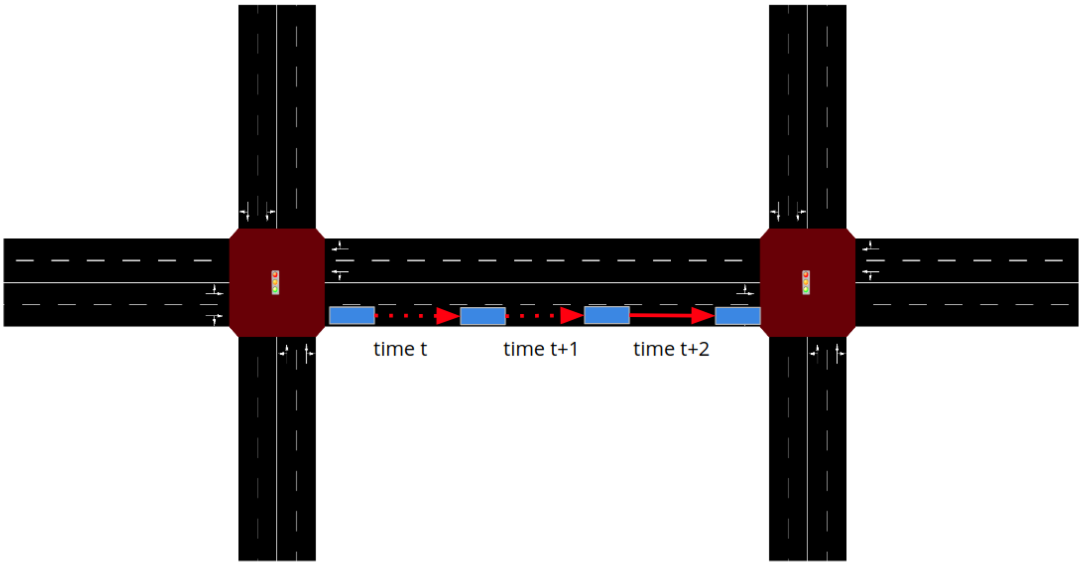

Figure 3.4: A vehicle needs time to travel across a road.

to the next junction will enter this part of the road).

In the shadow network, whenever a real vehicle enters the road network, a shadow vehicle is generated and enters the shadow network. Furthermore, one more shadow vehicle is generated with probability $\epsilon, 0 < \epsilon < 1$ and also enters the shadow network. This operation makes sure that the algorithm is stable, i.e., queue size will not go infinite (proper value and detail and Simulation Result and Analysis section of α show in Fig. 12) [18, 63].

When an actual vehicle goes into a road network from starting road R_i at t and wants to go to destination $d \in \mathbb{D}$, a shadow vehicle will also go into $\bar{B}_i^d(t)$. When that vehicle approaches the end part of road R_i, the shadow vehicle first leaves $\bar{B}_i^d(t)$ and then enters $\bar{Q}_i^d(t)$. We say a vehicle approaches the end part of one road if its speed is less than 5 km/h or it is within the range of 100 meters to the next junction.

Similarly, after an actual vehicle destined for destination $d \in \mathbb{D}$ leaves road R_i and goes into the adjacent road R_j at t, a shadow vehicle will leave $\bar{Q}_i^d(t)$ and goes into $\bar{B}_j^d(t)$. Fig. 4 shows vehicles need time to travel from one road to another road which depends on the vehicle speed and road length. Directly applying a back-pressure algorithm without traveling time may decrease the algorithm's performance. The movement of virtual shadow vehicles in the shadow network can be seen as control information exchange, based on which an agent performs its three tasks (details are given in the following section).

3.3.3 Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning

Our adaptive traffic control algorithm based on back-pressure and Q-Learning (ARD-BP-Q) is decentralized and the agent at each junction runs the following algorithm independently. At each time slot t, an agent performs the following three steps sequentially.

Step 1. Learning Global Congestion Information

The agent at a junction is responsible for estimating the route congestion level $R_{ij}^d(t)$ for all routes to destination d from road i and by the way of the neighbor road j. Each agent maintains a table $\boldsymbol{R}$ to store the value of $R_{ij}^d(t)$. At the beginning of each time slot, the agent exchanges information of the number of vehicles $\bar{Q}_j^d(t)$ at upstream roads around that junction and the table $\boldsymbol{R}$ with neighboring agents. After exchanging that information, the agent updates its route congestion estimate $R_{ij}^d(t)$ as follows:

$$R_{ij}^d(t) \leftarrow (1-\alpha)R_{ij}^d(t-1) + \alpha[\bar{Q}_j^d(t) + \gamma \min_k R_{jk}^d(t)] \tag{3.1}$$

where α is learning rate and γ is discount factor of Q-learning parameters, $0 < \alpha, \gamma <= 1$. If $R_{ij}^d > C_{max}$, set $R_{ij}^d = C_{max}$ where C_{max} is a positive constant. Bias quantity is a minimum value of estimating the route congestion level that starts from the origin to the destination, each agent then calculates a bias quantity $C_i^d(t)$ as follows:

$$C_i^d(t) = \min_j R_{ij}^d(t) \tag{3.2}$$

Finally, the bias quantity $C_i^d(t)$ will be used in Traffic Phase Selection.

Step 2. Traffic Phase Selection

The agents at each junction compute traffic pressure $w_{ij}^d(t)$ for all destinations and traffic movement, traffic pressure is the difference of queue length and bias quantity from the first road to the second road. Traffic pressure in our algorithm ARD-BP-Q (Algorithm 1) is defined as follows:

$$w_{ij}^d(t) = max\{(\bar{Q}_i^d(t) + C_i^d(t)) - (\bar{Q}_j^d(t) + C_j^d(t)), 0\} \tag{3.3}$$

Then the agent selects the destination d_{ij}^* that in return maximizes traffic pressure $w_{ij}^d(t)$ defined as follows:

$$d_{ij}^*(t) = \arg\max_d w_{ij}^d(t) \tag{3.4}$$

From the above equation, agents define $w_{ij}^{d_{ij}^*(t)}(t)$ as the weight of traffic movement which corresponds to one $d_{ij}^*(t)$ at time slot t.

Finally, the agent selects and activates the phase $p^{a*}(t) \in \mathbb{P}_a$ that releases the most traffic pressure defined as follows:

$$p^{a*}(t) = \arg\max_{p_l^a \in \mathbb{P}_a} \Sigma_{(R_i,R_j)\in p_l^a} w_{ij}^{d_{ij}^*(t)}(t) s_{ij}(t) \tag{3.5}$$

where s_{ij} is the number of vehicles that can move from road R_i to road R_j at time slot t.

Step 3. Vehicle Routing

A vehicle will follow the routing probabilities $P_{ij}^d(t)$ based on $\hat{\sigma}_{ij}^d(t)$ defined as follows:

$$P_{ij}^d(t) = \frac{\hat{\sigma}_{ij}^d(t)}{\Sigma_{k:\,(R_j,R_k)\in\mathbb{M}_a} \hat{\sigma}_{ik}^d(t)} \tag{3.6}$$

where $\hat{\sigma}_{ij}^d(t)$ is the estimated value of the expected number of shadow vehicles of destination d that moves from shadow queue $\bar{Q}_i^d(t)$ to shadow buffer $\bar{B}_j^d(t)$ which corresponds to road R_i and R_j. $\hat{\sigma}_{ij}^d(t)$ is updated by the agent of junction J_a for all destination $d \in \mathbb{D}$ and traffic movement $(R_i, R_j) \in \mathbb{M}_a$ as follows :

$$\hat{\sigma}_{ij}^d(t) = (1-\beta)\hat{\sigma}_{ij}^d(t-1) + \beta\sigma_{ij}^d(t) \tag{3.7}$$

where $0 < \beta < 1$. After a vehicle enters road R_i at time slot t. it will join lane L_{ij} with routing probability $P_{ij}^d(t)$.

Since our goal is to reduce vehicle traveling time, a heuristic is that we should let vehicles with a longer traveling time pass through a junction first. Thus, we also propose the following Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning with Vehicle traveling time (ARD-BP-QV Algorithm 2), which is the same as Algorithm 1 except that traffic pressure is defined as follows:

$$w_{ij}^d(t) = max\{(\bar{V}_i^d(t) + C_i^d(t)) - (\bar{V}_j^d(t) + C_j^d(t)), 0\} \tag{3.8}$$

where $\bar{V}_i^d(t)$ is the normalized value of the sum of the traveling time of vehicles in shadow queue $\bar{Q}_i^d(t)$, the normalized value is within range from 50-100. We need to normalize the vehicle traveling time to make it comparable to the quantity of bias $C_i^d(t)$ and $C_j^d(t)$.

3.4 Evaluation

In this section, we compare the performance of our algorithms with other algorithms below in an open-source simulator SUMO (Simulation of Urban MObility) [64]. Table 3.1 shows the compared algorithms in simulation. Our algorithms are ARD-BP-Q and ARD-BP-QV.

3.4.1 Simulation Setup

We implement a road network that mimics a simple grid road network and a real Stockholm road network under no roundabout or U-turns situations. The Stockholm road network was given by OpenStreetMap which can export the topology of a road network [65, 66].

Table 3.1: Compared algorithms in simulation.

Name	Detail
FC	Traffic signal control with fixed-cycles
SP-BP	Back-pressure and shortest path based traffic control algorithm [14]
AR-BP	Back-pressure based adaptive traffic signal control and vehicle routing without real-time control information update [18].
ARD-BP	Back-pressure based adaptive traffic signal control and vehicle routing with real-time control information update
ARD-BP-Q	Adaptive Traffic Control Algorithm Based on Back-Pressure and Global traffic information
ARD-BP-QV	Adaptive Traffic Control Algorithm Based on Back-Pressure and Global traffic information with Vehicle traveling time

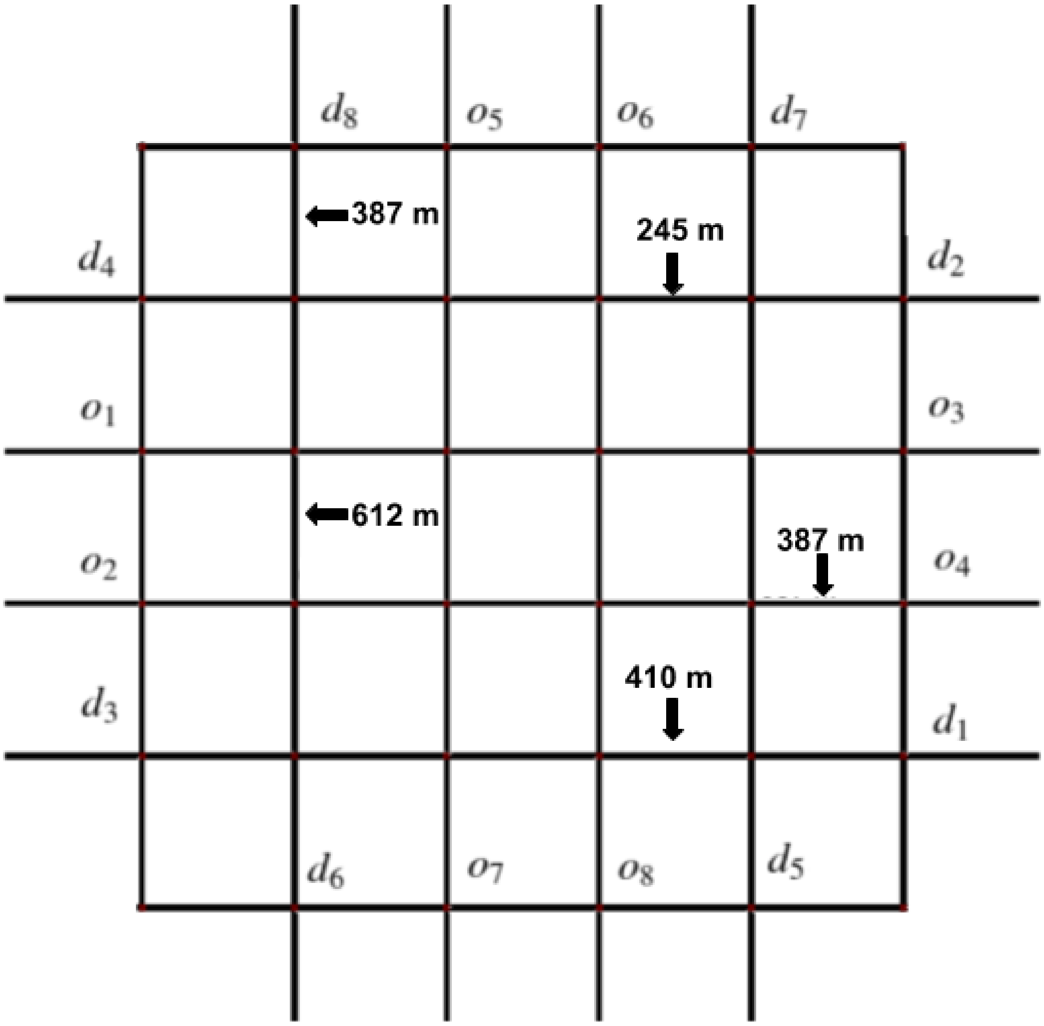

Figure 3.5: Grid road network structure that uses in SUMO with 8 pairs of origins and destinations.

Grid Road Network Scenario

The road map we used is given in Fig. 3.5. All roads have different lengths (250-950 meters) and speed limits (60-140 km/h). There are 8 origin and destination pairs $\{(o_1, d_1), (o_2, d_2), (o_3, d_3), ..., (o_8, d_8)\}$. All vehicles arrive at the starting roads with the same rates (360-2520 vehicles/hour) Duration of a slot is configured to be 15 seconds. Shadow vehicle generating probability ϵ is configured to be 0.02 and vehicle routing parameter β is configured to be 0.02.

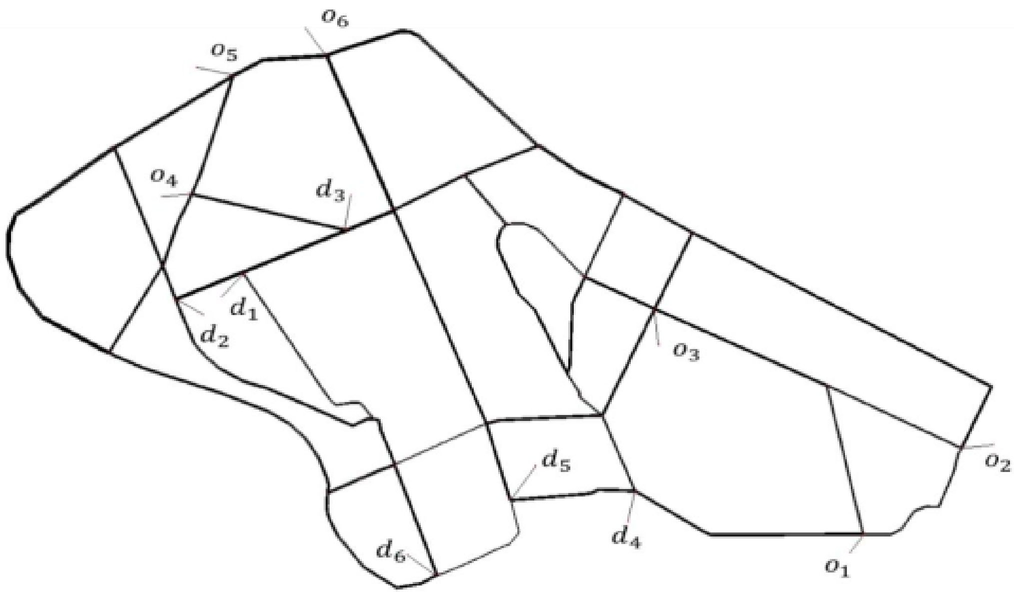

Figure 3.6: Road network structure of Stockholm city that uses in SUMO with 6 pairs of origins and destinations.

Stockholm Road Network Scenario

The road network consists of three and four way junctions as shown in Fig. 3.6. All roads have different lengths (400-1600 meters) and speed limits (60-140 km/h). Roads in this network are bi-directional. There are 6 pairs of origins and destinations $\{(o_1, d_1), (o_2, d_2), (o_3, d_3), ..., (o_6, d_6)\}$. All vehicles arrive at the starting roads with the same rates (360-2520 vehicles/hour). The duration of a slot is configured to be 15 seconds. Shadow vehicle generating probability ϵ is configured to be 0.02 and vehicle routing parameter β is configured to be 0.02.

3.4.2 Configuration

We define vehicle traveling time to be the time it takes a vehicle to travel from its starting road to its destination. For algorithms AR-BP, ARD-BP, ARD-BP-Q and ARD-BP-QV, parameter $\alpha = 2.5$ (Optimal value of α show in Fig. 14).

During simulations, we collect the following data: vehicle speed, number of vehicles in the road network, number of arriving vehicles at destinations, and a vehicle traveling time. Vehicle traveling time is the time it takes a vehicle to travel from its origin to its destination.

For algorithms FC and SP-BP, we simulate for 12200 seconds. We collect simulation data of vehicles that enter the road network before 7200 seconds only, because vehicles entering the road network after 7200 seconds may not arrive at their destinations.

For algorithms AR-BP, ARD-BP, ARD-BP-Q, and ARD-BP-QV, we simulate for 18200 seconds. We collect simulation data of vehicles that enter the road network from 6000-13200 seconds only because these algorithms need time to learn vehicle routing probabilities and reach a stable routing policy.

3.4.3 Results in Grid Road Network Scenario

In Fig. 3.7, our algorithm ARD-BP-QV achieves almost the lowest average traveling time under different vehicle arrival rates. Compared to ARD-BP, our algorithm ARD-BP-QV decreases the average vehicle traveling time by around 36%. Compared to ARD-BP-Q, algorithm ARD-BP-QV decreases average vehicle traveling time by around 12%. This indicates that the heuristic of letting vehicles with longer traveling time pass through the junction first is indeed an effective way to reduce vehicle traveling time.

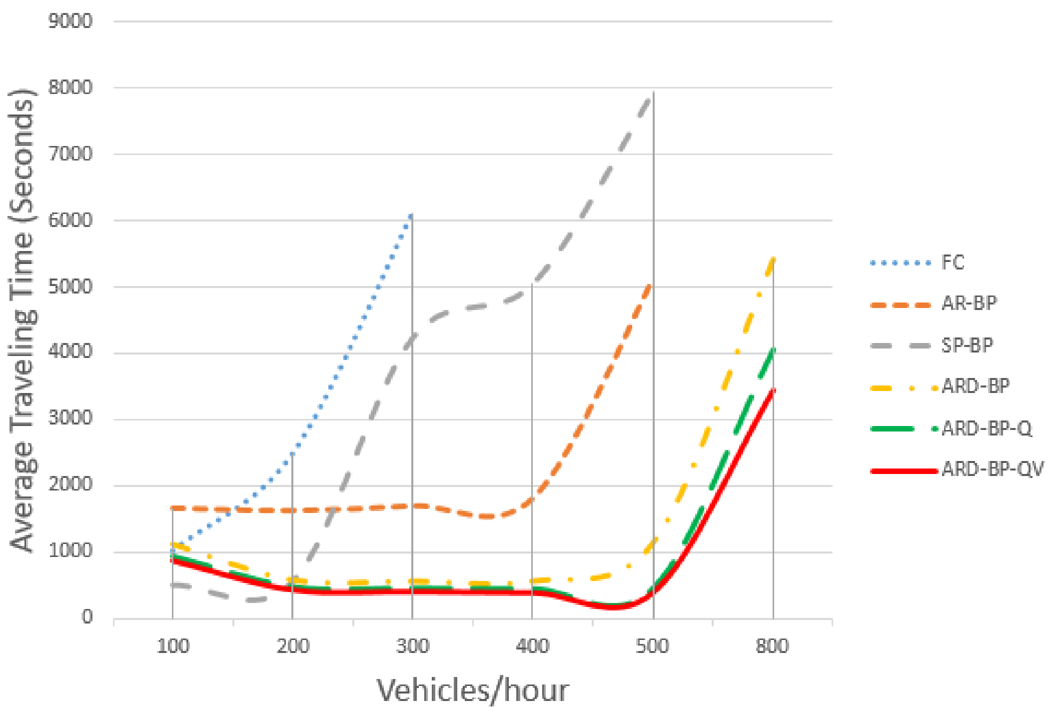

Figure 3.7: Average vehicle traveling time under different vehicle arrival rates.

Figure 3.8 shows simulation results of the average number of vehicles in a road network. This figure shows that the number of vehicles in the road network under the ARD-BP-QV algorithm is smaller than other algorithms, meaning less traffic congestion.

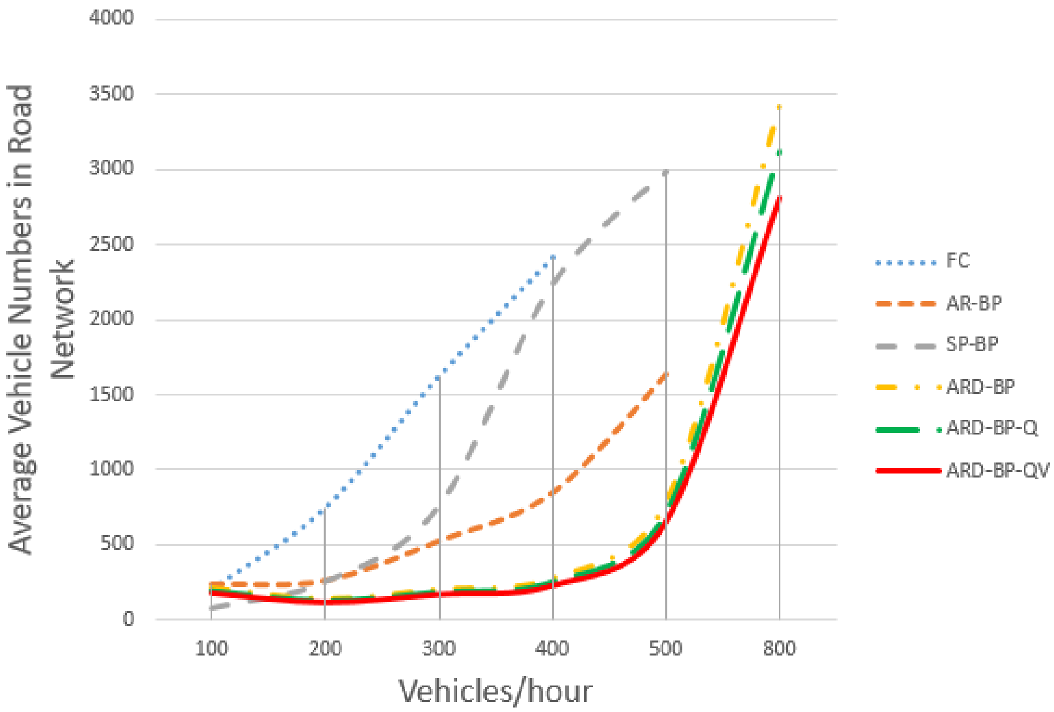

Figure 3.8: Average number of vehicles in the road network under different vehicle arrival rates.

Figure 3.9 shows that more vehicles can arrive at destinations under our algorithm ARD-BP-QV, meaning that more vehicles under other algorithms get stuck in the road network.

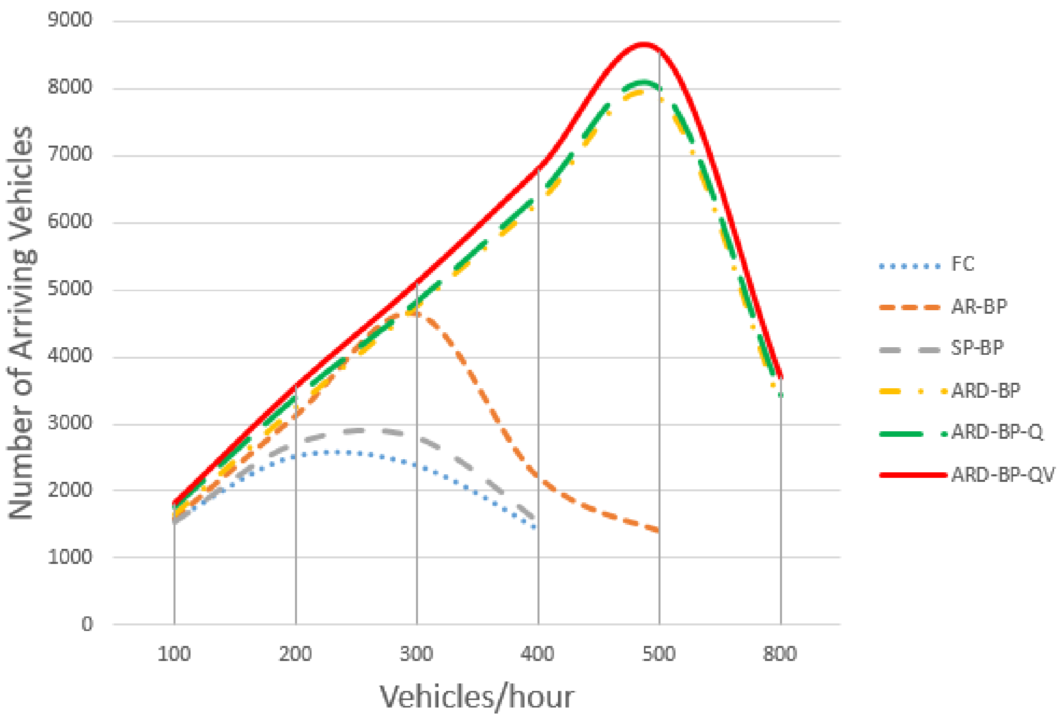

Figure 3.9: Number of vehicles arriving at destinations.

3.4.4 Results in Stockholm Road Network Scenario

Also in Fig. 3.10, our algorithm ARD-BP-QV achieves almost the lowest average traveling time under different vehicle arrival rates. Compared to ARD-BP, our algorithm ARD-BP-QV decreases the average vehicle traveling time by 17% to 37%. Compared to ARD-BP-Q, algorithm ARD-BP-QV decreases the average vehicle traveling time by 7% to 18%.

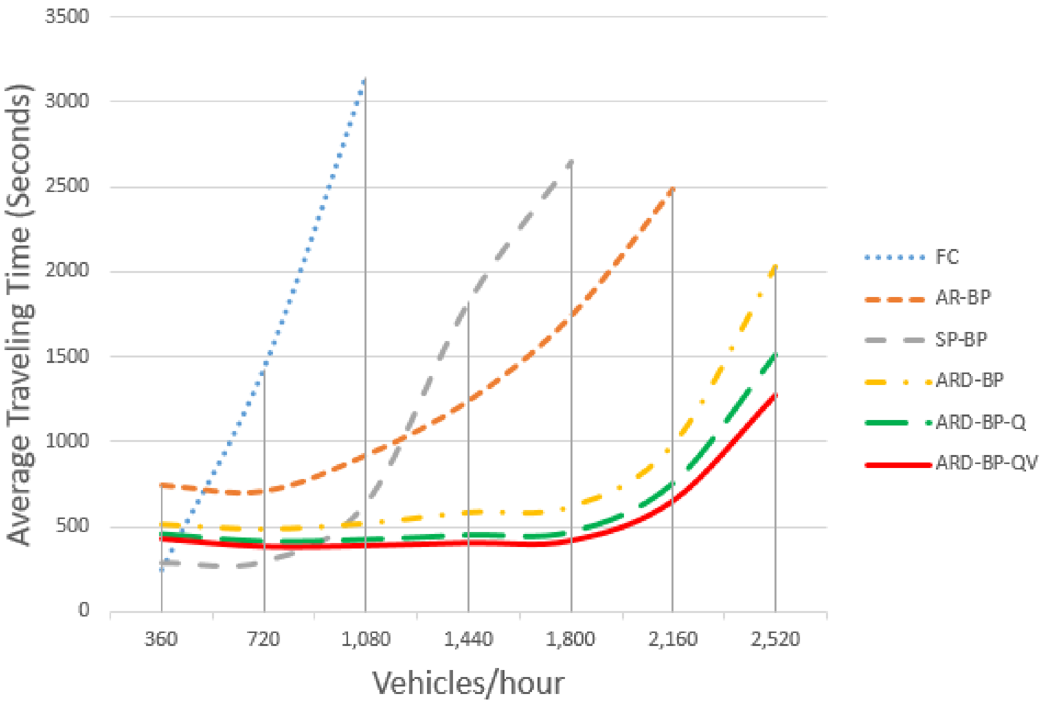

Figure 3.10: Average vehicle traveling time under different vehicle arrival rates.

Fig. 3.11 shows the simulation results of the average number of vehicles in the road network. This figure shows that the number of vehicles in the road network under the ARD-BP-QV algorithm is smaller than the other algorithms and also in the Stockholm road network scenario.

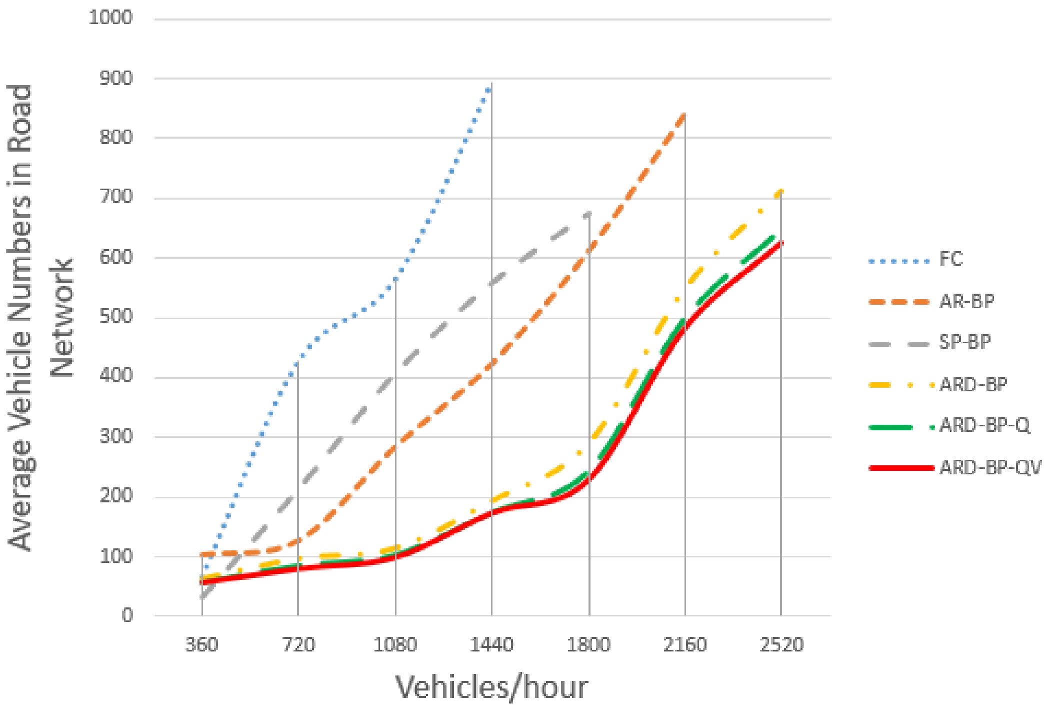

Figure 3.11: Average number of vehicles in the road network under different vehicle arrival rates.

Figure 3.12 shows that more vehicles can arrive at destinations under our algorithm ARD-BP-QV. Also in Fig. 3.12, more vehicles under other algorithms get stuck in the road network.

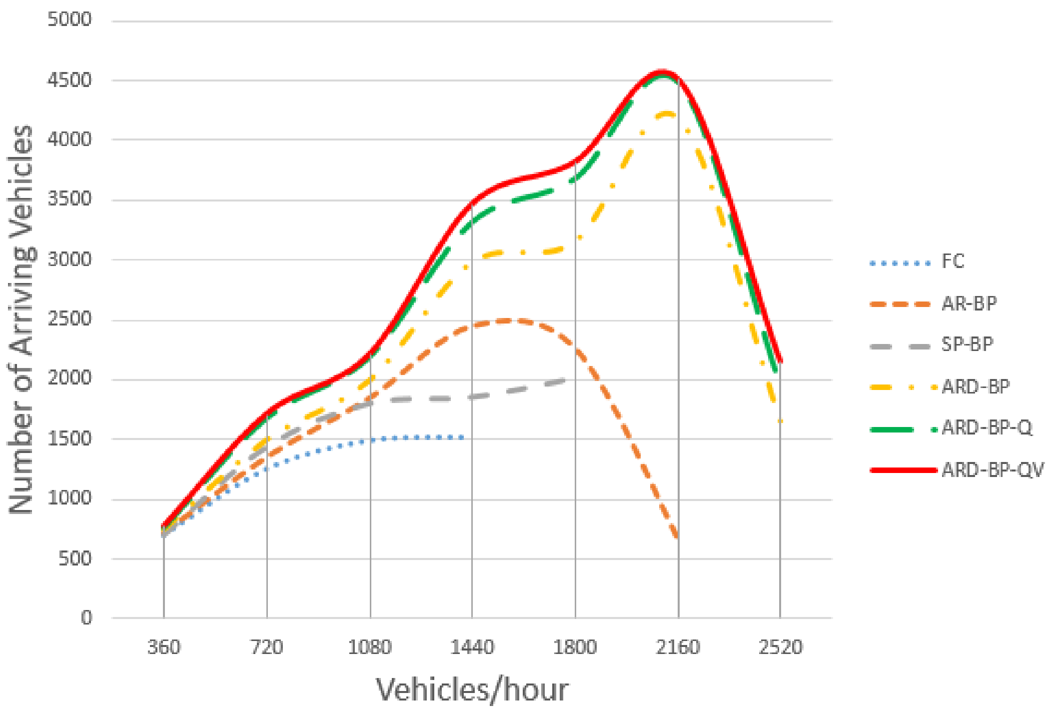

Figure 3.12: Number of vehicles arriving at destinations.

In the Stockholm road network scenario, we also evaluate the fairness of our algorithm. From Fig. 3.13, we see that most of the vehicles arrive at their destinations within 700 seconds, which is less than twice the average traveling time (385 seconds). So, our algorithm is fair for most vehicles.

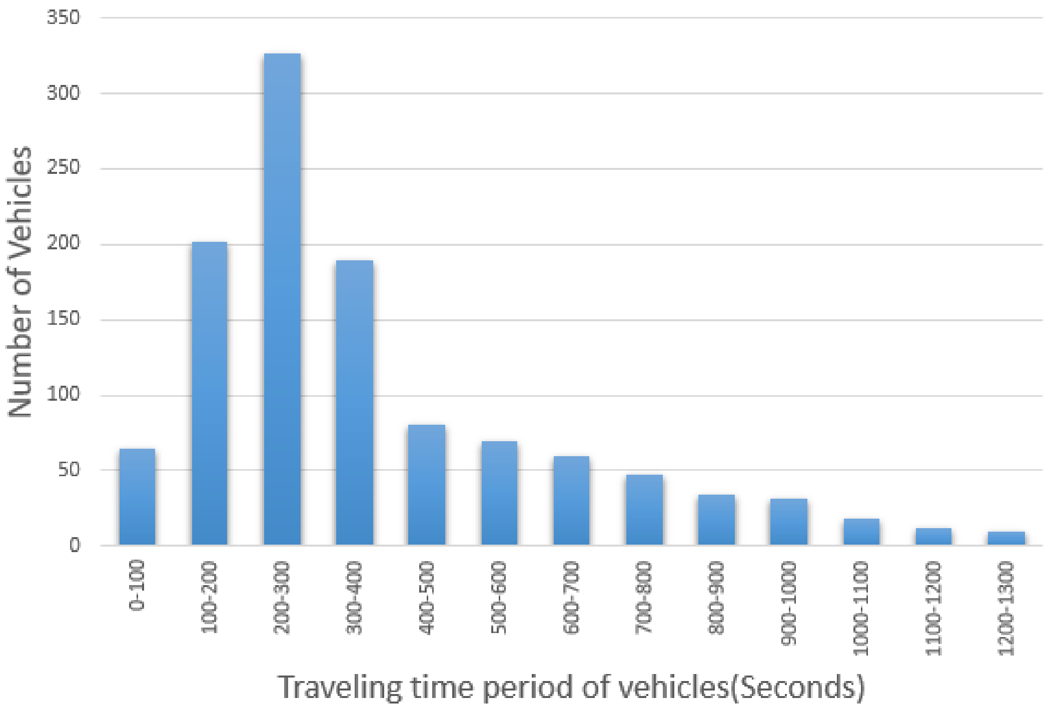

Figure 3.13: Histogram of the number of vehicles of different travelling times. Vehicle arrival rate is set to be 1080 vehicles/hour and the average traveling time is 385 seconds.

We also run simulations to check the impact of parameter α on ARD-BP-QV performance. As shown in Fig. 3.14. we need to properly set α in our algorithm to achieve the optimal performance.

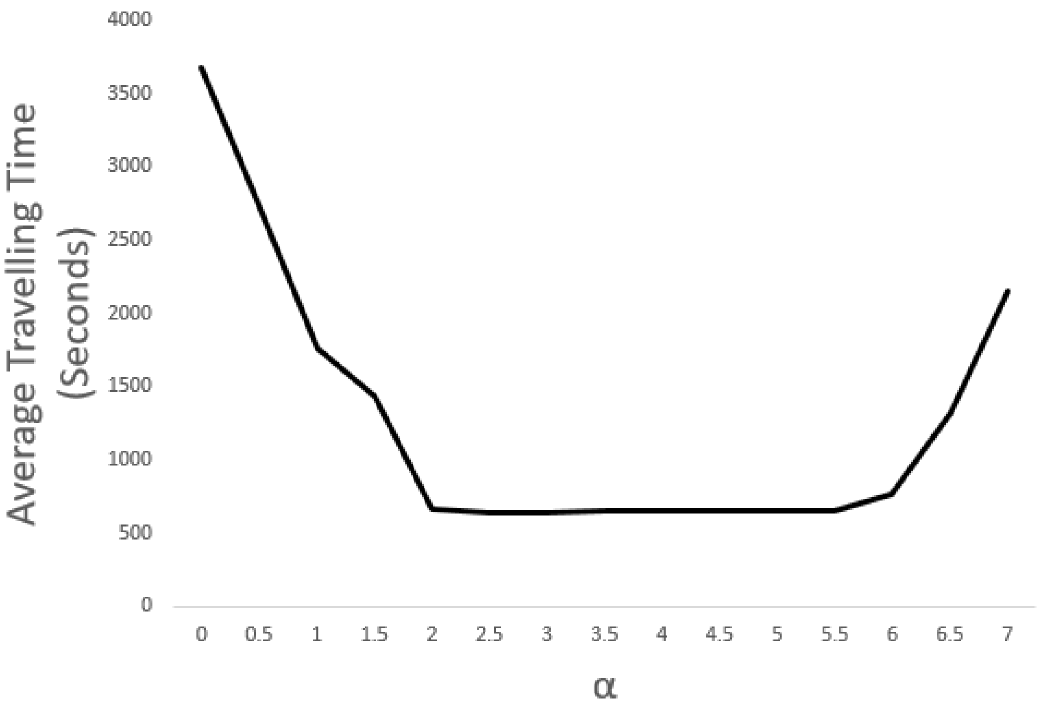

Figure 3.14: Performance under parameter α with rate of 450 vehicles/hour.

Finally, we simulate to check our algorithm under the Stockholm road network scenario with both self-driving or human driving vehicles, where all human-driving vehicles follow the shortest path route and the percentage of human-driving vehicles range from 10% to 60%. The simulation results are summarized in Fig. 3.15.

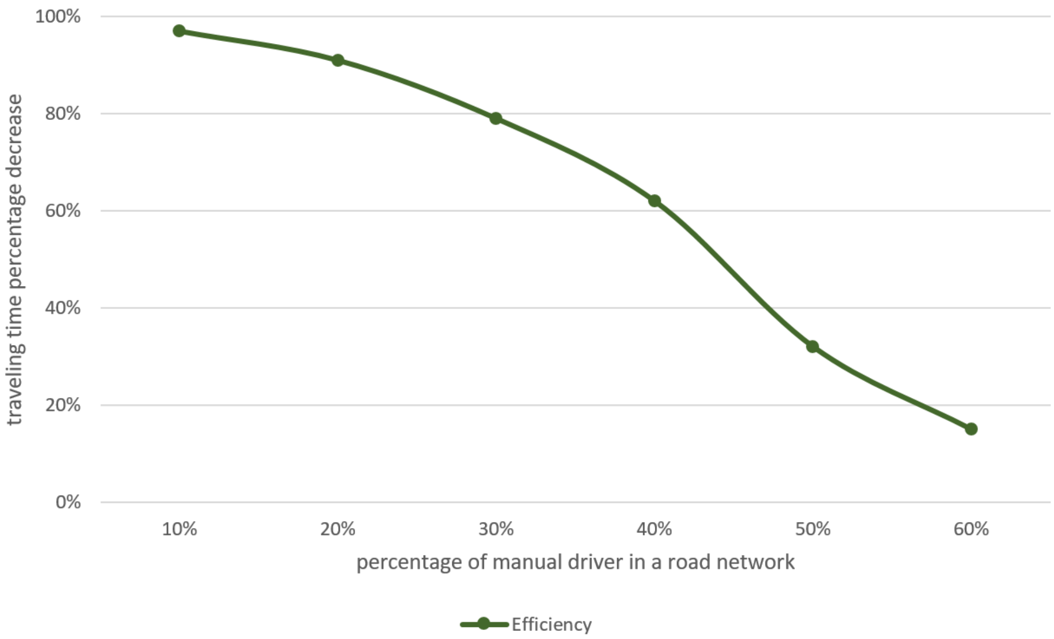

Figure 3.15: Vehicle traveling time of ARD-BP-QV under scenarios with both self-driving or human driving vehicles.

4 Conclusion

4.1 Summary

In this dissertation, we aim to measure traffic volume and control traffic lights using autonomous learning to reduce traffic congestion. To solve traffic congestion problems we provide a PAVEMENT, passing vehicle detection system with autonomous incremental learning using camera and vibration data and an adaptive traffic control algorithm based on back-pressure with global traffic information.

PAVEMENT, self-learning passing vehicle detection system with camera and vibration-based autonomous incremental learning. Initially, video-based passing vehicle detection is performed to get a ground truth label by using YOLOv3 + DeepSORT to detect and track vehicles from multiple frames of video. Then Vibration-based LDA and Incremental Learning are proposed for real-time autonomous learning. With the proposed system, the passing vehicle detection model using vibration data is automatically trained by setting the camera and the vibration sensor on the roadside. The trained models are used to detect passing vehicles without cameras once it is trained.

The experimental results demonstrate that the proposed method achieves comparable performance, 98.4% of accuracy and 98.0% F1-score, with the state-of-the-art method [28] which requires manual labeling while reducing the cost of manually labeling the ground truth, with only 15 times incremental learning steps with 1 minute data.

Since PAVEMENT provides us with a number of passing vehicles, we continue our research and proposed an adaptive traffic control algorithm based on back-pressure with global traffic information. Our algorithm controls traffic based on accurate real-time traffic information (achieved by using a shadow network) and global traffic information (achieved by using Q-learning). Our algorithm ARD-

BP-QV achieves almost the lowest average traveling time under different vehicle arrival rates. Compared to ARD-BP, our algorithm ARD-BP-QV decreases the average vehicle traveling time by 17% to 37%. Compared to ARD-BP-Q, algorithm ARD-BP-QV decreases the average vehicle traveling time by 7% to 18%.

4.2 Future works & Limitations

4.2.1 Embedded Systems

Through this study, in chapter 2 we implement LDA and incremental learning to detect passing vehicles in order to fulfill that approach embedded system is needed to implement our software into the embedded device.

4.2.2 Machine-to-Machine Communications

Through this study, in chapter 3 we assume all vehicles are self-driving vehicles which are amounted with accurate speed sensors and GPS devices, and can communicate in a timely way with control agents via networks, like vehicle-to-vehicle (V2V), vehicle-to-Road Side Unit (V2R), etc. The control agents are the computer programs placed at each junction to collect information of vehicle speed and vehicle position at every time slot for traffic control. A vehicle is also able to provide traveling time to control agents. The traveling time includes not only moving time but also waiting time to turn in junctions. To fulfill these conditions Machine-to-Machine (M2M) The main components of an M2M system include sensors, RFID, a Wi-Fi or cellular communications link, and autonomic computing software programmed to help a network device interpret data and make decisions. These M2M applications translate the data, which can trigger preprogrammed, automated actions.

4.2.3 Real-World Experiment

Since our method “adaptive traffic control algorithm based on back-pressure with Global traffic information” compute on simulation, vehicle accident or another event may happen in the road network which may affect the performance of the proposed algorithm.

References

[1] Lei, M., Lefloch, D., Gouton, P. and Madani, K.: A Video-Based Real-Time Vehicle Counting System Using Adaptive Background Method, *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pp. 523–528 (online), 10.1109/SITIS.2008.91 (2008).

[2] Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K. and Ohnishi, N.: A computer vision based vehicle detection and counting system, *2016 8th International Conference on Knowledge and Smart Technology (KST)*, pp. 224–227 (online), 10.1109/KST.2016.7440510 (2016).

[3] Crouzil, A., Khoudour, L., Valiere, P. and Cong, D. N. T.: Automatic vehicle counting system for traffic monitoring, *Journal of Electronic Imaging*, Vol. 25, No. 5, pp. 1 – 12 (online), 10.1117/1.JEI.25.5.051207 (2016).

[4] Oltean, G., Florea, C., Orghidan, R. and Oltean, V.: Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation, *2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pp. 240–243 (2019).

[5] Mohana and Aradhya, H. R.: Object Detection and Tracking using Deep Learning and Artificial Intelligence for Video Surveillance Applications, *International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 12 (online), 10.14569/IJACSA.2019.0101269 (2019).

[6] Bathija, A. and Sharma, G.: Visual object detection and tracking using Yolo and sort, *Int. J. Eng. Res. Technol.(IJERT)*, Vol. 8, pp. 705–708 (2019).

[7] Li, L., Wen, D. and Yao, D.: A Survey of Traffic Control with Vehicular Communications, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, pp. 425–432 (2014).

[8] Ahmed, E. and Gharavi, H.: Cooperative Vehicular Networking: A Survey, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 3, pp. 996–1014 (2018).

[9] Hodges, S., Taylor, S., Villar, N., Scott, J., Bial, D. and Fischer, P. T.: Prototyping Connected Devices for the Internet of Things, *IEEE Computer*, Vol. 46, No. 2, pp. 26–34 (2013).

[10] Robertson, D. I. and Bretherton, R. D.: Optimizing Networks of Traffic Signals in Real-Time the SCOOT Method, *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, pp. 11–15 (1991).

[11] Sims, A. G. and Dobinso, K. W.: The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits, *IEEE Transactions on Vehicular Technology*, Vol. 29, No. 2, pp. 130–137 (1980).

[12] Lowrie, P. R.: SCATS, Sydney Co-ordinated Adaptive Traffic System: A Traffic Responsive Method for Controlling Urban Traffic, *Road and Traffic Authority of New South Wales* (1990).

[13] Le, T., Kovács, P., Walton, N., Vu, H. L., Andrew, L. L. H. and Hoogendoorn, S. S. P.: Decentralized Signal Control for Urban Road Networks, *Transportation Research Part C: Emerging Technologies*, Vol. 58, Part C, pp. 431–450 (2015).

[14] Wongpiromsarn, T., Uthaicharoenpong, T., Wang, Y., Frazzoli, E. and Wang, D.: Distributed Traffic Signal Control for Maximum Network Throughput, *Proceedings of the 15th IEEE International Conference on Intelligent Transportation Systems (ITSC 2012)*, pp. 588–595 (2012).

[15] Gregoire, J., Frazzoli, E. and de La Fortelle Tichakorn Wongpiromsarn, A.: Back-Pressure Traffic Signal Control with Unknown Routing Rates, *IFAC Proceedings Volumes*, Vol. 47, No. 3, pp. 11332–11337 (2014).

[16] Gregoire, J., Qian, X., Frazzoli, E., de La Fortelle, A. and Wongpiromsarn, T.: Capacity-Aware Backpressure Traffic Signal Control, *IEEE Transactions on Control of Network Systems*, Vol. 2, No. 2, pp. 164–173 (2015).

[17] Kulcsár, B., Ampountolas, K. and Dabiri, A.: Single-Region Robust Perimeter Traffic Flow Control, *Proceedings of the 14th European Control Conference (ECC 2015)*, pp. 2628–2633 (2015).

[18] Zaidi, A. A., Kulcsár, B. and Wymeersch, H.: Back-Pressure Traffic Signal Control With Fixed and Adaptive Routing for Urban Vehicular Networks, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 8, pp. 2134–2143 (2016).

[19] Shobha, B. S. and Deepu, R.: A Review on Video Based Vehicle Detection, Recognition and Tracking, *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, pp. 183–186 (online), 10.1109/CSITSS.2018.8768743 (2018).

[20] Wei, Y., Tian, Q., Guo, J., Huang, W. and Cao, J.: Multi-vehicle detection algorithm through combining Harr and HOG features, *Mathematics and Computers in Simulation*, Vol. 155, pp. 130–145 (online), https://doi.org/10.1016/j.matcom.2017.12.011 (2019). International Conference on Mathematical Modeling and Computational Methods in Science and Engineering.

[21] Alpatov, B. A., Babayan, P. V. and Ershov, M. D.: Vehicle detection and counting system for real-time traffic surveillance, *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4 (online), 10.1109/MECO.2018.8406017 (2018).

[22] Choudhury, S., Chattopadhyay, S. P. and Hazra, T. K.: Vehicle detection and counting using haar feature-based classifier, *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, pp. 106–109 (online), 10.1109/IEMECON.2017.8079571 (2017).

[23] Memon, S., Bhatti, S., Thebo, L. A., Talpur, M. M. B. and Memon, M. A.: A video based vehicle detection, counting and classification system, *International Journal of Image, Graphics and Signal Processing*, Vol. 11, No. 9, p. 34 (2018).

[24] Wu, H., Zhang, X., Story, B. and Rajan, D.: Accurate Vehicle Detection Using Multi-camera Data Fusion and Machine Learning, *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3767–3771 (online), 10.1109/ICASSP.2019.8683350 (2019).

[25] Zhang, Y., Wang, J. and Yang, X.: Real-time vehicle detection and tracking in video based on faster R-CNN, *Journal of Physics: Conference Series*, Vol. 887, p. 012068 (online), 10.1088/1742-6596/887/1/012068 (2017).

[26] Taylor, C. J., Furuya, T., Kuhn, B., Shah, M. and Pendharkar, A.: Vehicle Detection and Counting at Road Intersections Using Video Data (2018).

[27] Bhavke, A. and Pai, S.: Advance automatic toll collection amp; vehicle detection during collision using RFID, *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, pp. 1–5 (online), 10.1109/ICNTE.2017.7947958 (2017).

[28] Yoshida, M., Akiyama, S., Moriyama, Y., Takeshima, Y., Kondo, Y., Suwa, H. and Yasumoto, K.: Traffic Census Sensor Using Vibration Caused by Passing Vehicles, *Sens. Mater.*, Vol. 33, No. 1, pp. 1–16 (2021).

[29] Yonar, F.: Piezoelectric Devices in Traffic Information Collection (2019).

[30] González, B., Jiménez, F. J. and De Frutos, J.: A Virtual Instrument for Road Vehicle Classification Based on Piezoelectric Transducers, *Sensors*, Vol. 20, No. 16 (online), 10.3390/s20164597 (2020).

[31] Puan, O. C., Nor, N. S. M., Mashros, N. and Hainin, M. R.: Applicability of an automatic pneumatic–tube–based traffic counting device for collecting data under mixed traffic, *IOP Conference Series: Earth and Environmental Science*, Vol. 365, p. 012032 (online), 10.1088/1755-1315/365/1/012032 (2019).

[32] Burnos, P., Gajda, J., Piwowar, P., Sroka, R., Stencel, M. and Zeglen, T.: Measurements of Road Traffic Parameters Using Inductive Loops and Piezoelectric Sensors, *Metrology and Measurement Systems*, Vol. 14, p. 187–203 (2007).

[33] Dong, H., Wang, X., Zhang, C., He, R., Jia, L. and Qin, Y.: Improved Robust Vehicle Detection and Identification Based on Single Magnetic Sensor, *IEEE Access*, Vol. 6, pp. 5247–5255 (online), 10.1109/ACCESS.2018.2791446 (2018).

[34] Wang, Q., Zheng, J., Xu, H., Xu, B. and Chen, R.: Roadside Magnetic Sensor System for Vehicle Detection in Urban Environments, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 5, pp. 1365–1374 (online), 10.1109/TITS.2017.2723908 (2018).

[35] Taghvaeeyan, S. and Rajamani, R.: Portable roadside sensors for vehicle counting, classification, and speed measurement, *IEEE Intelligent Transportation Systems Magazine*, Vol. 15, No. 1, pp. 73–83 (online), 10.1109/TITS.2013.2273876 (2014).

[36] Zhu, H. and Yu, F.: A Cross-Correlation Technique for Vehicle Detections in Wireless Magnetic Sensor Network, *IEEE Sensors Journal*, Vol. 16, pp. 4484–4494 (2016).

[37] Odat, E., Shamma, J. and Claudel, C.: Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 5, pp. 1593–1606 (online), 10.1109/TITS.2017.2727224 (2018). Publisher Copyright: © 2000-2011 IEEE. Copyright: Copyright 2018 Elsevier B.V., All rights reserved.

[38] Bao, X., Zhan, Y., Xu, C., Hu, K., Zheng, C. and Wang, Y.: A novel dual microwave Doppler radar based vehicle detection sensor for parking lot occupancy detection, *IEICE Electronics Express*, Vol. advpub, p. 13.20161087 (online), 10.1587/elex.13.20161087 (2016).

[39] Luo, Z., Habibi, S. and v. Mohrenschildt, M.: LiDAR Based Real Time Multiple Vehicle Detection and Tracking, *International Journal of Computer and Information Engineering*, Vol. 10, No. 6, pp. 1125 – 1132 (online), https://publications.waset.org/vol/114 (2016).

[40] Fang, J., Meng, H., Zhang, H. and Wang, X.: A Low-cost Vehicle Detection and Classification System based on Unmodulated Continuous-wave Radar, pp. 715 – 720 (online), 10.1109/ITSC.2007.4357739 (2007).

[41] Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A. and Wang, Y.: Review of Road Traffic Control Strategies, *Proceedings of the IEEE*, Vol. 91, No. 12, pp. 2043–2067 (2003).

[42] Chen, X.-F. and Shi, Z.-K.: Real-Coded Genetic Algorithm for Signal Timing Optimization of a Single Intersection, *Proceedings of the 1st International Conference on Machine Learning and Cybernetics (ICMLC 2002)* (2002).

[43] Niittymäki, J.: Installation and Experiences of Field Testing a Fuzzy Signal Controller, *European Journal of Operational Research*, Vol. 131, No. 2, pp. 273–281 (2001).

[44] Pappis, E. and Mamdani, C. P.: A Fuzzy Logic Controller for a Traffic Junction, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, No. 10, pp. 707–717 (1977).

[45] Pillac, V., Gendreau, M., Guŕret, C. and Medaglia, A. L.: A Review of Dynamic Vehicle Routing Problems, *European Journal of Operational Research*, Vol. 225, No. 1, pp. 1–11 (2013).

[46] Kenyon, A. S. and Morton, D. P.: Stochastic Vehicle Routing with Random Travel Times, *Transportation Science*, Vol. 37, No. 1, pp. 69–82 (2003).

[47] Secomandi, N. and Margot, F.: Reoptimization Approaches for the Vehicle-Routing Problem with Stochastic Demands, *Operations Research*, Vol. 57, No. 1, pp. 214–230 (2008).

[48] Thomas, B. W. and White, C. C.: Anticipatory Route Selection, *Transportation Science*, Vol. 38, No. 4 (2004).

[49] Thomas, B. W.: Waiting Strategies for Anticipating Service Requests from Known Customer Locations, *Transportation Science*, Vol. 41, No. 3, pp. 319–331 (2007).

[50] Novoa, C. and Storer, R.: An Approximate Dynamic Programming Approach for the Vehicle Routing Problem with Stochastic Demands, *European Journal of Operational Research*, Vol. 196, No. 2, pp. 509–515 (2009).

[51] Chai, H., M.Zhang, H., Ghosal, D. and Chuah, C.-N.: Dynamic Traffic Routing in a Network with Adaptive Signal Control, *Transportation Research Part C: Emerging Technologies*, Vol. 85, pp. 64–85 (2017).

[52] Kim, S., Lewis, M. E. and White, C. C.: Optimal Vehicle Routing with Real-Time Traffic Information, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 2, pp. 178–188 (2005).

[53] Redmon, J. and Farhadi, A.: YOLOv3: An Incremental Improvement, *arXiv* (2018).

[54] Duan, C. and Li, X.: Multi-target Tracking Based on Deep Sort in Traffic Scene, *Journal of Physics: Conference Series*, Vol. 1952, No. 2, p. 022074 (online), 10.1088/1742-6596/1952/2/022074 (2021).

[55] Wang, J., Sun, X. and Guo, J.: A Region Tracking-Based Vehicle Detection Algorithm in Nighttime Traffic Scenes, *Sensors*, Vol. 13, No. 12, pp. 16474–16493 (online), 10.3390/s131216474 (2013).

[56] Cai, Z., Fan, Q., Feris, R. S. and Vasconcelos, N.: A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection, *CoRR*, Vol. abs/1607.07155 (online), http://arxiv.org/abs/1607.07155 (2016).

[57] Pham, T.-A. and Yoo, M.: Nighttime Vehicle Detection and Tracking with Occlusion Handling by Pairing Headlights and Taillights, *Applied Sciences*, Vol. 10, No. 11 (online), 10.3390/app10113986 (2020).

[58] Kuang, H., Zhang, X., Li, Y.-J., Chan, L. and Yan, H.: Nighttime vehicle detection based on bio-inspired image enhancement and weighted score-level feature fusion, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 4, pp. 927–936 (online), 10.1109/TITS.2016.2598192 (2017).

[59] O'Malley, R., Glavin, M. and Jones, E.: Vehicle Detection at Night Based on Tail-Light Detection, (online), 10.4108/ICST.ISVCS2008.3546 (2010).

[60] Sarikan, S. S. and Ozbayoglu, A. M.: Anomaly Detection in Vehicle Traffic with Image Processing and Machine Learning, *Procedia Computer Science*, Vol. 140, pp. 64–69 (online), https://doi.org/10.1016/j.procs.2018.10.293 (2018). Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018.

[61] Liu, Y., Gao, J. and Ito, M.: Back-Pressure Based Adaptive Traffic Signal Control and Vehicle Routing with Real-Time Control Information Update, *Proceedings of the 20th IEEE International Conference on Vehicular Electronics and Safety (ICVES 2018)*, pp. 1–6 (2018).

[62] Maipradit, A., Gao, J., Kawakami, T. and Ito, M.: Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning, *Proceedings of the 22nd IEEE Intelligent Transportation Systems Conference (ITSC 2019)*, pp. 1995–1999 (2019).

[63] Athanasopoulou, E., Bui, L. X., Ji, T., Srikant, R. and Stolyar, A.: Back-Pressure-Based Packet-by-Packet Adaptive Routing in Communication Networks, *IEEE/ACM Transactions on Networking*, Vol. 21, No. 2, pp. 244–257 (2012).

[64] Krajzewicz, D., Hertkorn, G., Rössel, C. and Wagner, C. P.: SUMO (Simulation of Urban MObility)—An Open-Source Traffic Simulation, *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, pp. 183–187 (2002).

[65] Haklay, M. and Weber, P.: OpenStreetMap: User-Generated Street Maps, *IEEE Pervasive Computing*, Vol. 7, No. 4, pp. 12–18 (2008).

[66] Haklay, M.: How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets, *Environment and Planning B: Urban Analytics and City Science*, Vol. 37, No. 4, pp. 682–703 (2010).

Publication List

Peer Review Journal

[1] **Arnan Maipradit**, Tomoya Kawakami, Ying Liu, Juntao Gao, and Minuro Ito, "An Adaptive Traffic Signal Control Scheme Based on Back-pressure with Global Information," *Journal of Information Processing*, 2021, vol. 29, pp. 124-131, February 2021

International Conference (Thesis-related)

[1] **A. Maipradit**, J. Gao, T. Kawakami and M. Ito, "Adaptive Traffic Control Algorithm Based on Back-Pressure and Q-Learning," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1995-1999.

[2] **A. Maipradit**, Y. Moriyama, T. Okuro, M. Yoshida, N. Tachimori, H. Suwa, K. Yasumoto, "PAVEMENT: Passing Vehicle Detection System with Autonomous Incremental Learning using Camera and Vibration Data, to appear in IEEE VTC 2022-Fall.