Article

Estimation of Lane-Level Traffic Flow by Using Deep Learning Technique

Chieh-Min Liu [1] and Jyh-Ching Juang [2]

[1] Department of Electrical Engineering, National Cheng-Kung University; jeremyckt@gmail.com
[2] Department of Electrical Engineering, National Cheng-Kung University; juang@mail.ncku.edu.tw

Featured Application: This paper proposed an object detection and tracking system which can count vehicles, estimate velocity of vehicles, provide estimation of traffic flow for applying on traffic monitoring and control.

Abstract: This paper proposes a neural network which fuses the data received from a camera system on a gantry, to detect moving objects and calculate relative position and velocity of the vehicles traveling on a freeway, this information is used to estimate the traffic flow. To estimate the traffic flow at both microscopic and macroscopic view, this paper used YOLO v4 and DeepSORT for vehicle detection and tracking, then counting the number of vehicles pass through the freeway by drawing virtual lines and hot zones, also counting the velocity of each vehicle. The information is then pass to the traffic control center, in order to monitoring and control traffic flow on freeways, and analyzing freeway conditions.

Keywords: traffic flow; object detection; object tracking; deep learning

1. Introduction

Self-Driving Cars is one of the main developments of artificial intelligence. By combining the data from internal and external sensors and detecting the surrounding objects accurately is helpful for driving safety. Object detection on computer vision has been developed in recent years, including detecting surrounding vehicles, pedestrians, obstacles and measuring the distances associated with them, recognizing traffic signals and signs for self-driving cars, and counting amounts of vehicles passing through highway. Fully-Connected Network (FCN) techniques to detect and localize objects as 3D boxes from range scan data has been developed.

This paper focuses on various methods of object detection and tracking for vehicles on highway, which was applied on estimating regional traffic flow, detecting objects and measuring their position and velocity. To achieve it, a neural network will be proposed, which combined with processes of moving object detection and calculation of relative position and velocity of moving objects, to estimate surrounding traffic flow. By identifying three lanes in each direction (northbound and southbound) individually, classifying vehicles and counting their amounts, vehicle density of each traffic lane can be calculated correctly. Such new method that we proposed would give real-time accurate data for traffic control and analysis.

In Taiwan, traffic jam during commuting hours is a tremendous problem for such a highly populated and urbanized small island. The main purpose for this paper is to provide data to Traffic Control Center to reduce the traffic jam by applying artificial intelligence technique. Existing models for vehicles detection and tracking might have repeated counting due to inconsistency of object tracking. To solve this problem, our model introduced "virtual lines" and "hot zones" (detailed in Section 3.3.2) to accurately count the number of vehicles passing through each lane of the section of freeway. The benefit of adding hot zone is that when a vehicle moved close to the virtual line, it could be tracked more accurately, and avoid repeated count. A deep learning procedure by neural network technique will be designed for estimating velocity of vehicles. By providing these data obtained from our system to Traffic Control Center, it could be beneficial for recommending optimal route to each car, self-driving or not, and improve the traffic situation.

This paper has three main features. First, the traffic flow counted by Freeway Bureau, Ministry of Transportation and Communications (MOTC), Taiwan, used Electronic Toll Collection (ETC) sensor to count cars, but it would miss some cars that didn't install the ETC. In this paper, object detection and tracking were used to count vehicles and estimate velocity, which could not only solve the problem, but also reduce costs. Second, traffic flow of each lane was calculated individually instead of simply two directions in this paper. Third, our technique could show different colors on each lane according to its velocity level. These features would be beneficial to the traffic flow monitoring and control by Freeway Bureau, MOTC, Taiwan, especially with situations of heavy traffic, such as commuting hours and holidays.

2. Related Works

Nowadays, deep learning has been a popular academic and technological topic, a popular method applied on some AI techniques, including application on self-driving cars and transportation. Traffic flow monitoring on video was applied on different traffic scenes, such as city roads, highways, intersections. [1, 2] The research of deep learning applied on solving traffic flow prediction was also developed in recent years. In the 21st century, researchers proposed a deep learning based traffic flow prediction method used to identify various cars running on highway. For example, the software tools developed by Y. LV et al used stack autoencoder (SAE) approach to represent traffic flow features maps for prediction. The greedy layerwise unsupervised learning algorithm was applied to pretrain the network, followed by updating the model's parameters by a fine-tuning process to improve the prediction. [3]

There are various other deep learning models developed to predict traffic flow. For example, N.G. Polson et al proposed a deep learning model developed to predict short-term traffic flow. [4] W.S. McCulloch developed a model with a combination of a linear model with L1 regularization and some tanh layers. [5] V. Minh et al designed a deep learning predictor which takes an input vector and output via different layers. [6] They develop a linear model by using previous measurements to provide a forecast.

The techniques for object detection and object tracking are important applications for vehicles counting. For object detection, there are object detectors such as YOLO, Faster R-CNN, SSD, etc. YOLO is a state-of-the-art, real-time object detection system. [7, 8] This object detection method proposed by Joseph Redmon et al used regression methods to calculate the bounding box and related classification probability of the detected objects. A single neural network is used to predict the bounding box and classification probability directly from the complete image in one evaluation. Since the entire detection is a single network, the end-to-end detection performance can be directly optimized. [7] YOLO could solve the special regression problem of object detection frames, and could identify the object and predict bounding box simultaneously. Therefore, YOLO has become one of the most widely-used object detection tool, which is especially powerful in high real-time detection tasks, and newer versions have been developed to improve detection performances. YOLO v2 not only has higher accuracy, but also has higher efficiency, by adding batch normalization layer in the convolutional neural networks, and has high-resolution classifier. YOLO v3 uses multi-label classification, in which an object may belong to multiple categories at the same time. YOLO v3 replaces the softmax function with an independent logistic function to calculate the probability that the input belongs to a specific label. In addition, instead of using the mean square error to calculate the classification loss, YOLO v3 uses the two-class cross-entropy loss for each category. This approach also reduces the computational complexity brought by the softmax function. YOLO v3 uses DarkNet-53 as the main backbone of convolutional neural networks to extract features from the input image, which has better efficiency and detection performance than other backbones. [9] The architecture of DarkNet-53 is shown in Table 1.

Table 1. The architecture of DarkNet-53 [9]

Type	Filters	Size	Output	
Convolutional	32	3×3	256×256	
Convolutional	64	3×3/2	128×128	
Convolutional	32	1×1		
Convolutional	64	3×3		1×
Residual			128×128	
Convolutional	128	3×3/2	64×64	
Convolutional	64	1×1		
Convolutional	128	3×3		2×
Residual			64×64	
Convolutional	256	3×3/2	32×32	
Convolutional	128	1×1		
Convolutional	256	3×3		8×
Residual			32×32	
Convolutional	512	3×3/2	16×16	
Convolutional	256	1×1		
Convolutional	512	3×3		8×
Residual			16×16	
Convolutional	1024	3×3/2	8×8	
Convolutional	512	1×1		
Convolutional	1024	3×3		4×
Residual			8×8	
Avgpool		Global		
Connected		1000		
Softmax				

YOLO v4 was released in April 2020, which has a higher accuracy and efficiency for object detection, and also reduced the hardware requirements. [10] YOLO v4 established an efficient and powerful object detection model, verified the influence of the state-of-art Bag-of-Freebies and Bag-of-Specials object detection methods in the training process, and improved the techniques of optimizing network during training and testing. YOLO v4 uses CSPDarknet53, the combination of Darknet53 and Cross Stage Partial Network (CSP-Net), [11] as the main backbone of convolutional neural networks.

The procedure of object tracking generally included the following steps. [12] First, input the video and execute the object detectors such as YOLO, Faster R-CNN or SSD, to detect the objects and obtain their detection frames. Then, obtain all the corresponding targets from the detection frames and extract their features, including apparent features and/or motion features. Afterwards, calculate the similarity between these objects in adjacent frames. Finally, after linking the same objects together, assign different objects with different IDs.

For object tracking, SORT (Simple Online and Realtime Tracking) is combined with Kalman filter and Hungarian Algorithm, which enhanced the efficiency of multiple object tracking. The Kalman filter algorithm is divided into two processes, prediction and update. The algorithm defines the motion state of the target as 8 normally distributed vectors. When the target moves, the target frame position and speed of the current frame are predicted based on the target frame and speed of the previous frame. The predicted value and the observed value, the two normal distribution states are linearly weighted, and the current state of the system prediction is obtained. In the main step of multiple object tracking, the similarity is calculated and the similarity matrix of the two frames before and after is obtained. The Hungarian algorithm solves the real matching goal of the two frames before and after solving this similarity matrix. [13]

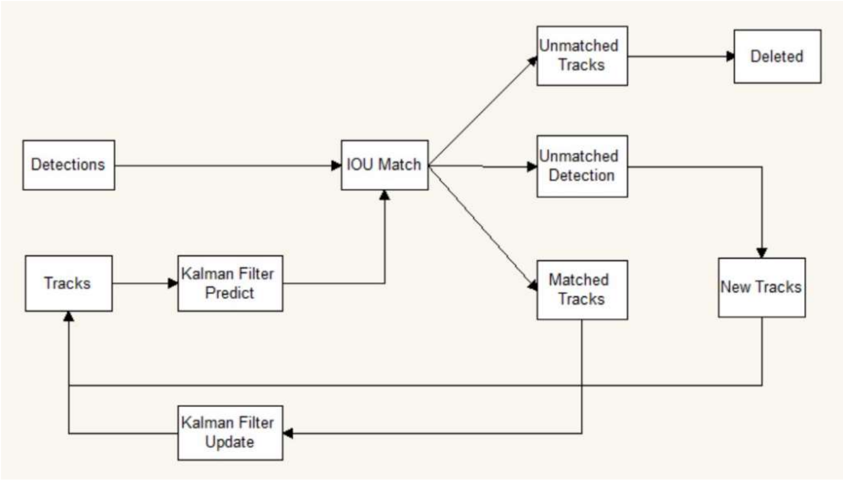

Figure 1. Object tracking procedure of SORT

DeepSORT is developed from SORT, which added the information of exterior of the objects, to match the objects in adjacent frames. The procedure of DeepSORT is that, first, it uses Kalman filter predicts trajectory, then uses the Hungarian algorithm to match the predicted trajectory with the detections in the current frame (cascade matching and IOU matching), and then update the Kalman filter. DeepSORT can be applied on Multiple Object Tracking (MOT), which can assign each vehicles with different IDs.

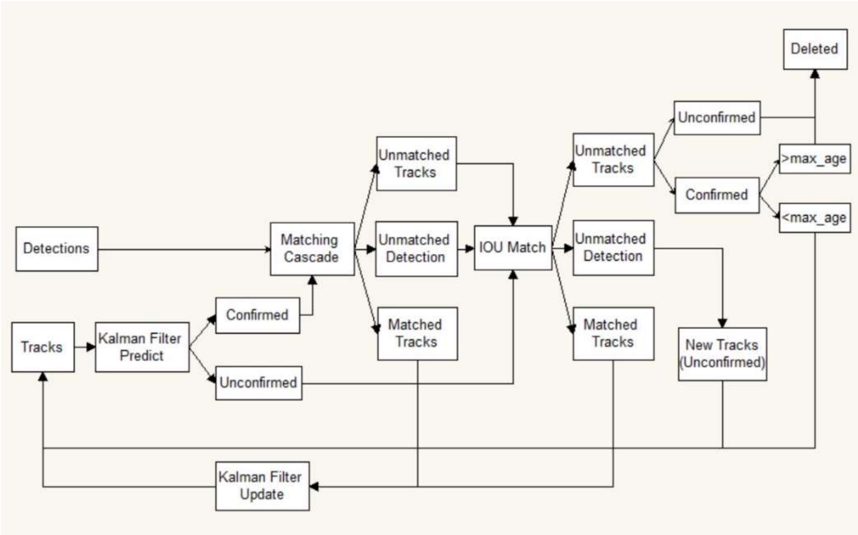

Figure 2. Object tracking procedure of DeepSORT

Z. Wang et al proposed a multiple object tracking system, which has a detection model for target localization and an appearance embedding model for data association. [14] A. Fedorov et al applied faster R-CNN detector and SORT tracker on traffic flow estimation with data from camera. [15] The application of YOLO and DeepSORT can be used on many multiple object tracking scenarios, not only for traffic, [16] but also for crowd control, [17] and for detecting and tracking moving obstacles. [18]

This paper applied artificial intelligence on improving the method of monitoring traffic flow. Traffic flow prediction has been one of the main topics for self-driving cars, and some related methods and algorithms have been developed in the past decade to improve route guidance for cars and assist traffic management and control. Neural network models were also developed to applied on traffic flow prediction. By identifying vehicles from traffic videos with artificial neural networks (ANNs), results of our paper could make effective contribution to the improvement of highway traffic jam situations in Taiwan.

3. Methods

3.1. Process and Flow Chart

The whole process shown in the flow chart (Fig.3). After inputting video, our system detects the vehicles by YOLO v4, and then tracks the vehicles by DeepSORT. Finally, the number of vehicles that passed through each lane and the velocity of each vehicle was calculated.

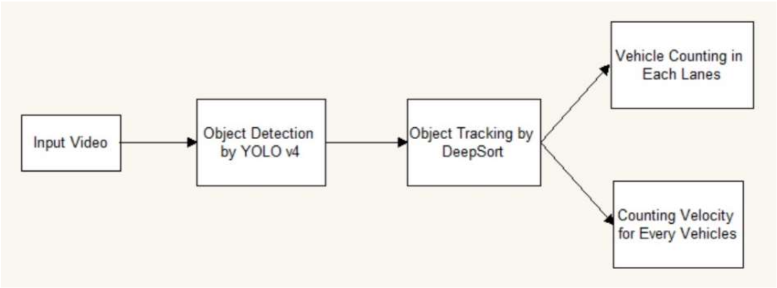

Figure 3. Flow chart of the whole process

3.2. Traffic Flow Calculation Model

Here are some fundamental quantities for traffic flow: [19] Density ρ is defined as the number of vehicles per unit length. The speed and positions of the n-th vehicle are denoted as v_n and r_n, respectively. Flow rate f is the number of vehicles passing a fixed position per unit time (say one hour). Time mean speed is the average vehicle speeds in the fixed position over time, space mean speed is the average vehicle speeds over a hot zone in a fixed time. Bulk velocity is defined as the ratio of flow rate to density, i.e. $u = \frac{f}{\rho}$.

The traffic flow could be described microscopically and macroscopically, as proposed by B. Seibold. [19] The microscopic description of traffic flow was used to characterize each vehicles' behavior, apply on many car-following models (e.g., the intelligent driving model), and facilitate a self-driving car's decision of keeping at the same lane or switching to other lanes, to avoid backing up.

For microscopic description, there are individual trajectories, vehicle position equation $p_a(t) = (p_{aX}(t), p_{aY}(t)), a = 1, 2, ..., n$, vehicle velocity equation $v_1(t)$, $v_2(t)$, …, $v_n(t)$ and acceleration equation $a_a(t) = \dot{v}_a(t) = \frac{dv_a(t)}{dt}$. [19]

For microscopic traffic models, there are two models needed to setup. "Follow the leader" model: Accelerate/decelerate towards the velocity of the vehicle ahead of the self-driving car itself. If the position and velocity of the vehicle ahead is p_2 and v_2, then the acceleration according to this model should be:

$$a_F = \frac{v_2 - v}{p_2 - p}, \tag{2}$$

"Optimal velocity" model: Accelerate/decelerate towards an optimal velocity that depends on your distance to the vehicle ahead. Then the acceleration according to this model should be:

$$a_O = V(p_2 - p) - v, \tag{3}$$

By combining both models, the acceleration of self-driving car could be obtained in the following equation:

$$a = \alpha a_F + \beta a_O = \alpha \frac{v_2 - v}{p_2 - p} + \beta(V(p_2 - p) - v), \tag{4}$$

On the other hand, the macroscopic description of traffic flow is usually lane-aggregated, but multi-lane models can also be formulated. The macroscopic description of traffic flow is also good for estimation and prediction, and for mathematical analysis of emergent features. It can examine bigger ranges of highway, judging the traffic in each section is convenient or heavy, and assists navigation and determining the path planning for a car.

Macroscopic description are based on field quantities, including vehicle density $\rho(p,t)$, velocity field $u(p,t)$, and flow-rate field $f(p,t) = \rho(p,t)u(p,t)$, where p is position on road and t is time. [19]

To set up macroscopic traffic models, the following equations are needed. The number of vehicles in the section interval *c* to *d* could be obtained from vehicle density $\rho(p,t)$ as follow:

$$m(t) = \int_c^d \rho(p,t)dp, \tag{5}$$

Traffic flow rate (flux):

$$f(p,t) = \rho(p,t)u(p,t), \tag{6}$$

Change of numbers of vehicles equals inflow f(c) minus outflow f(d) dynamically as follow:

$$\frac{d}{dt}m(t) = \int_c^d \rho_t dp = f(c) - f(d) = -\int_c^d f_p dp, \tag{7}$$

This equation holds for any choice of c and d:

$$\rho_t + (\rho u)_p = 0, \tag{8}$$

The microscopic view of traffic flow is studied in this paper, and the macroscopic view of traffic flow will be studied in future. Both descriptions have their own importance and usage.

3.3 Object Identification and Vehicle Tracking

3.3.1 Pretrain and Retrain for Object Identification

Our inputs were obtained from the MP4 videos taken at several fixed points on the National Freeway No.1 from Freeway Bureau, MOTC in Taiwan. In the process of object detection, we used YOLO v4 to identify objects in every frame (approximately 2000 to 9000 frames per video). The model was pre-trained by Coco dataset, and then retrained with our own picture data of classified vehicles, whose categories corresponded to Coco dataset. [9, 20] In each frame, the bounding-boxes of all vehicles were labelled and classified. In addition, the centers of each vehicle were pointed out and the distances between them were measured. By visual inspection of the outcome, these vehicles were correctly identified and well-classified into cars, buses, and trucks, the three relevant categories of vehicles for the Taiwanese freeway.

3.3.2 Object Tracking by Drawing Virtual Lines and Hot Zones

In this paper, we used DeepSORT to track the objects (vehicles) identified by YOLO v4. In short, DeepSORT first uses Kalman filter to predict the trajectories of objects from previous frames, then uses the Hungarian algorithm to match the predicted trajectories with the detected objects in the current frame (using cascade matching and IOU matching), and then updates the Kalman filter for further predictions. [13] By executing DeepSORT, we could calculate the similarities between the apparent features and/or motion features of objects in adjacent frames, link the same objects together, assign different IDs to different objects, and track the objects throughout the video.

In order to calculate lane-specific traffic flows, we drew a virtual line on each direction of the highway (Fig.4) in the images and count the number of vehicles (as defined by the centers of their bounding-boxes) passing such lines in each lane. As a result, a vehicle that changed lane in the video would only be counted on one of the lanes, determined by its location while passing through the virtual line. There are 6 lanes on the sections of National Freeway No.1 where the videos were taken (Fig.5). The input video's resolution is 1920*1080 pixels, with 30 frames per second.

During object tracking, sometimes the system would lose track of an object for a few frames and then recaptured again, and sometimes it might be identified as a different object and reassigned with a different ID. Such inconsistency of object tracking might cause instability of the system such as repeated counting of vehicles and confusion in speed calculation (in Section 3.3.3). Therefore, we modified the algorithm by adding hot zones in the vicinity of virtual lines (Fig.4), and only performed object tracking when the center is within the hot zones. The introduction of hot zones improved the accuracy of vehicle tracking and subsequent calculations.

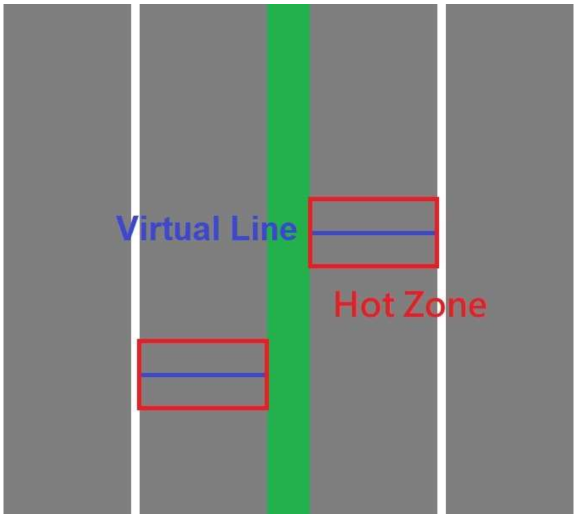

Figure 4. Virtual lines and hot zones of the inner lanes of both directions. (conceptualization)

Figure 5. Sample image of the video taken on National Freeway No.1.

3.3.3 Estimation of Velocities of Each Vehicles

As shown in Figure 6, when drawing two virtual lines separated by a fixed distance (e.g. 20m) in a bigger hot zone and obtaining the two time points when a vehicle passes through the two virtual lines (t_1 and t_2) by the aforementioned object tracking technique, the velocity of the vehicle can be calculated as:

$$v = \frac{d}{t_2 - t_1}, \tag{9}$$

where d is the fixed distance.

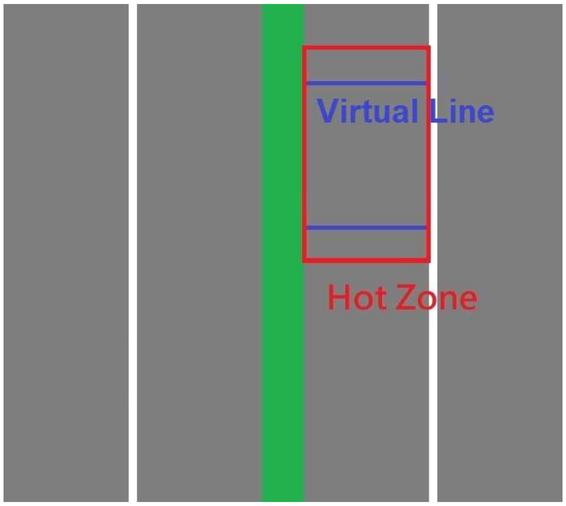

Figure 6. Two virtual lines and hot zone for velocity calculation. (conceptualization)

The parameters can be acquired as follows:
$p_1, p_2, \ldots, p_n$: The positions of vehicles
$v_1, v_2, \ldots, v_n$: The velocities of vehicles

After inputting these data, the velocities of vehicles would be classified into 5 levels as defined by Freeway Bureau, MOTC, Taiwan (Table 2):

Table 2. Levels of speed classification.

Levels	Speed Range
Level 1	0 km/hr $\leq v_a <$ 20 km/hr, a=1,2,…,n

Level 2	20 km/hr $\leq v_a < 40$ km/hr, a=1,2,…,n
Level 3	40 km/hr $\leq v_a < 60$ km/hr, a=1,2,…,n
Level 4	60 km/hr $\leq v_a < 80$ km/hr, a=1,2,…,n
Level 5	$v_a \geqq 80$ km/hr, a=1,2,…,n

Similarly, we could classify each lane into different speed levels from the average speed of vehicles, and color-code the lanes for visualization.

4. Experiments

The results from digital image processing, object detection, and object tracking were shown as follows. We obtained accurate vehicle counts for both the northbound and southbound directions, and for each lane separately. Moreover, the velocities of all vehicles were estimated.

4.1 Digital Image Processing

The video taken at National Freeway No.1 will be processed into a MP4 video as the input for our system.

4.2 Object Detection Results

We had shown that our model, using YOLO v4, trained by Coco dataset and our own picture dataset, could identify all the vehicles on the freeway and perform accurate classifications (Fig.7).

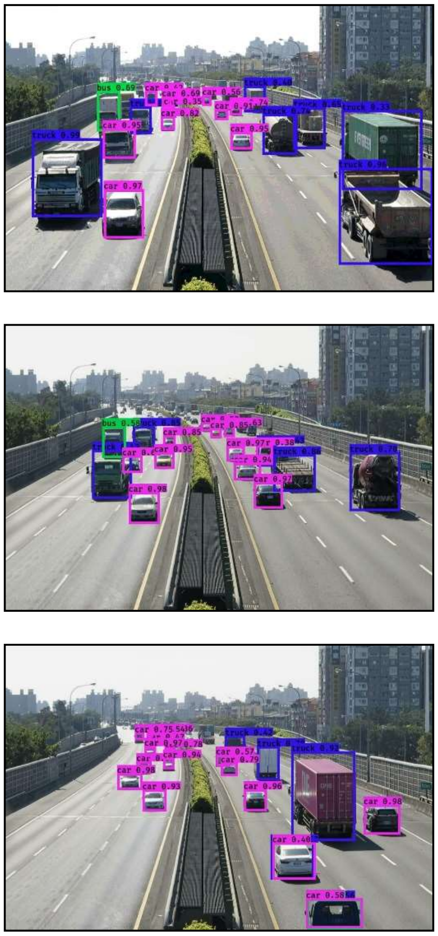

Figure 7. Object detection results on the video taken on National Freeway No.1.

The identified vehicles were wrapped with their respective bounding boxes. The colors of the bounding boxes represent the categories of the vehicles: cars (magenta), trucks (blue), and buses (green).

4.3 Vehicle Counting on Both Northbound and Southbound Directions

By adding virtual lines and hot zones on both the northbound and southbound sides of the freeway, the vehicles entered in the hot zones could be tracked well, and thus we could correctly count the number of vehicles passing through the virtual line (Fig.8), which in turn would yield the flow rate (the number of vehicles passing through per unit time) of the section of the freeway in any given time interval.

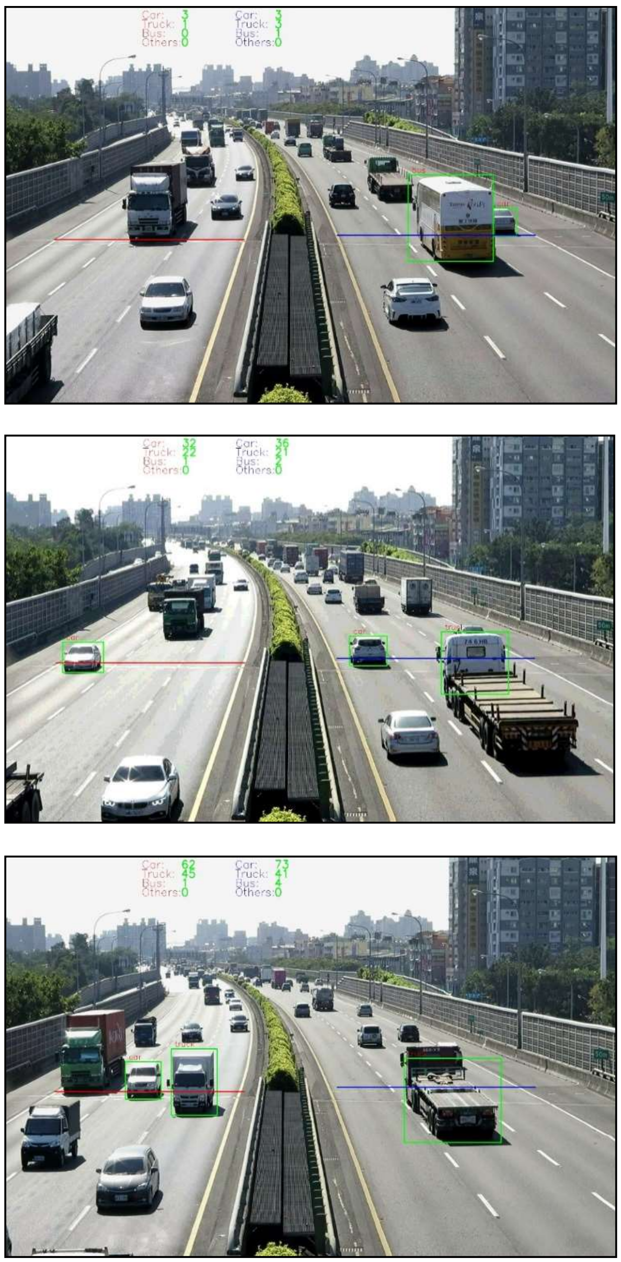

Figure 8. The results of vehicle counting on both northbound and southbound on the video taken on National Freeway No.1.

The red line represents the virtual line over the northbound direction, while the blue line represents the virtual line over the southbound direction. Note that only the vehicles approaching the virtual lines were tracked (marked with green boxes).

4.4 Vehicle Counting in Each Lane on Both Northbound and Southbound Directions

The hot zones and virtual lines we introduced also allowed us to calculate the vehicle count and the flow rate of each lane individually (Fig.9). Each vehicle would only be counted once, in the lane where it passed through the virtual line.

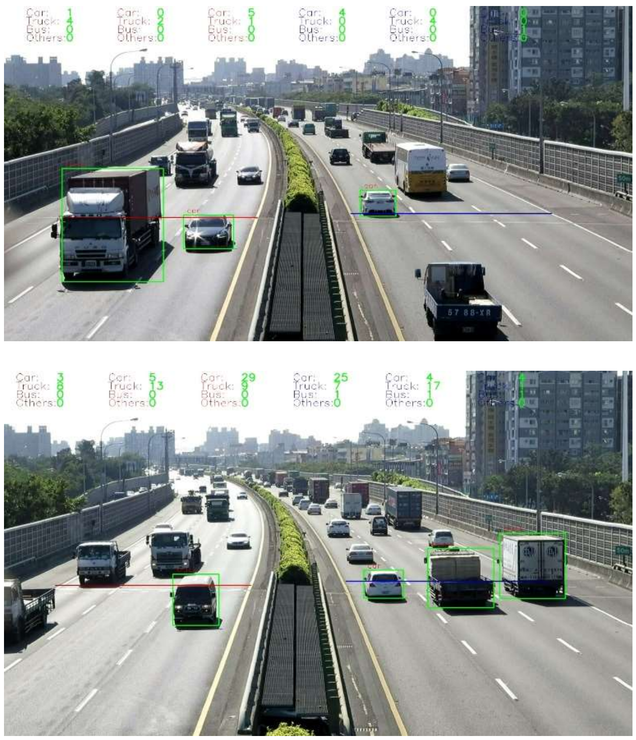

Figure 9. The results of vehicle counting on both northbound and southbound on the video taken on National Freeway No.1.

The red line represents the connected virtual lines over the three lanes of the northbound direction, while the blue line represents the connected virtual lines over the three lanes of the southbound direction. The green boxes mark the vehicles tracked in the hot zones.

The vehicle counting results were illustrated in Table 3. We compared the vehicle counts on each lane by visual inspection and by object detection of our system in 3 minutes. The errors were caused by double-labeling large trucks (e.g., trailer trucks, container cars), erroneously counting vehicles to the adjacent lane (Lane 2 to Lane 1, Lane 5 to Lane 6), and large trucks blocking small cars from the camera.

Table 3. Vehicle Counting Results (in 3 minutes).

Northbound (toward the camera)								
Lane 1			**Lane 2**			**Lane 3**		
Actual	Detected	Error	Actual	Detected	Error	Actual	Detected	Error
44	56	27.3%	60	59	1.7%	61	76	24.6%

Southbound (away from the camera)								
Lane 4			**Lane 5**			**Lane 6**		
Actual	Detected	Error	Actual	Detected	Error	Actual	Detected	Error
70	78	11.4%	70	68	2.9%	50	51	2%

4.5 Velocity Estimation

The velocity of each vehicle on the freeway was calculated by equation (9).

Figure 10. Velocity estimation on the video taken on National Freeway No.1

4.6 Velocity Level

As the velocities of the vehicles passing through the lane were calculated, the color of the virtual line changes to represent the velocity level of the lane according to Table 2. The colors representing Level 1 to 5 are purple, red, orange, yellow and green, respectively.

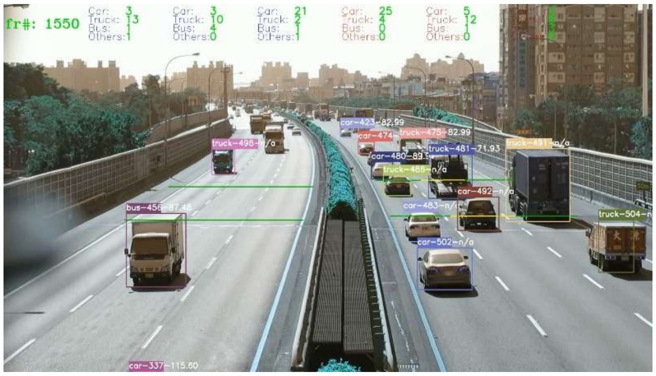

Figure 11. Velocity level of each lanes showing on the video taken on National Freeway No.1

5. Discussion

In this paper, we trained a YOLO v4 model for object detection, which successfully identified all vehicles in the input video, and classified them into cars, buses, and trucks. Afterwards, we used DeepSORT for multi-object tracking, which could track these vehicles frame-to-frame and trace their individual trajectories.

Furthermore, we introduced virtual lines and hot zones into our DeepSORT-based tracking system, which improved the accuracy of lane-specific vehicle counts (mostly by avoiding duplicates), and facilitated the calculation of individual vehicle velocities. Such improvements in turn enabled accurate estimation of lane-specific flow rate and average speed. The visualization of color-coded velocity level on each lane could show the condition of each lane.

There are several potential applications for our system. First of all, if our system is linked to the Traffic Control Center, we can obtain real-time information such as numbers of different types of vehicles, lane-specific flow rate, lane-specific average speed, etc., which can provide detailed real-time monitoring and analysis of traffic conditions throughout the highway systems, and play an important part in the decision-making or even law-enforcing process for the authorities. In Taiwan, video cameras are set at most sections in our National Highway System. Therefore, our system could be implemented at a much lower cost and provide more accurate and detailed traffic monitoring than the current ETC-based monitoring system by Freeway Bureau, MOTC, which can only provide the overall flow rate of ETC-enabled vehicles. While Freeway Bureau, MOTC spent more funds on assembling ETC for vehicles counting, our object detection and tracking system can simply implement on camera, which could not only save funds for vehicles counting, but also enhance system functions by added function of vehicle speed estimation. Since our system is less costly and requires less maintenance than ETC-based or induction-based traffic monitoring system, it would also be more feasible for developing countries that lack adequate infrastructure.

In addition, the real-time information acquired from our system could be transmitted to individual drivers or self-driving cars, to help them make decisions such as staying in the same lane or switching to another lane if approaching to a partially congested section of the freeway. When integrated with navigation systems, it would be helpful for recommending optimal route to each vehicle.

Moreover, the abundant data for individual vehicles obtained by our system, such as their types, speeds, timings of passing a section, and the lanes used, would be helpful for the analysis of road occupants' habit and could be used to build up traffic models and prediction models if our system were set up throughout the entire freeway.

In this paper, we further simplified the classifications of detected objects to cars, buses, and trucks (any other object would be classified as others), because these are the only three types of vehicles that would appear on a Taiwanese freeway. In fact, it is well within YOLO's ability to provide more detailed and broader range of classifications. When coupled with our real-time lane-specific traffic statistics, this extended system would be capable of not only detect congestion, but also help identify the cause of congestion (e.g., if an animal appeared on the freeway, or if a scooter illegally entered the freeway).

The paper is concerned with vehicle detection, vehicle tracking, and speed estimation. The proposed approach is improved from other existing approaches in the following ways. In our literature review with applications of YOLO and DeepSORT, [15-18] two were related to traffic flow estimation. A. Federov et al. only counted vehicles passing the crossroad with different directions, [15] and A. Santos et al. only counted vehicles in Brazilian roads. [16] In this paper, we not only counted vehicles passing through the freeway in each lane, but also calculated the velocity of these vehicles, and the velocity level in each lane. Therefore, our system has more functional application for traffic monitoring and control. In addition, our system used YOLO v4 for object detection, which has higher accuracy and efficiency than YOLO v3 which A. Santos et al. used. Our system used DeepSORT for object tracking, which added Deep Association Metric on SORT and enabled assigning objects with different IDs, and thus can perform Multiple Object Tracking (MOT) better than SORT, which A. Federov et al. used.

Because we defined the position of a vehicle by the center of its bounding box, if the curvature of the road surface is too large or the size of the vehicle is too large, sometimes that center point would fall in the adjacent lane, and the vehicle would be erroneously counted in the adjacent lane. This problem only affected the lane-specific estimations, but not the estimations for the overall directions.

6. Conclusions & Future Work

In recent years, deep learning techniques for optimizing object detection and tracking were developed rapidly, and have been applied to human and vehicle traffic flow. [16, 17] To our knowledge, this paper was the first to introduce virtual lines and hot zones to a deep learning object detection and tracking system for lane-specific vehicle count and velocity estimation.

Our contributions to traffic flow estimation detailed in the 3rd to 5th paragraph of the "Discussion" section. Briefly, the contributions are: The system proposed in this paper can provide detailed real-time information including the number of vehicles passes through in each lane of the section of freeway, these vehicles' types, and the velocities of these vehicles. Such information can be provided to the authorities for traffic monitoring and control, shared with individual vehicles (including self-driving cars) for navigation, or analyzed by official or academic institutions for traffic modelling and prediction. Since the system proposed in this paper could be simply implemented on video cameras set on the freeway, its efficiency, low cost, and high data quality would be favored over ETC-based traffic monitoring system for traffic counting and analysis.

In the future, we will keep on improving the prediction accuracy, especially that of lane-specific counting. We will also study on applying the techniques on macro view of traffic flow and hope to develop an attention model which can provide the whole traffic situation in larger scale automatically.

All detection schemes are subject to cybersecurity attacks. The data that are used in the paper are from an authorized organization (MOTC) which is believed to be less vulnerable. In the future, this issue will be further investigated.

Supplementary Materials: The following are available online at www.mdpi.com/xxx/s1, Figure S1: title, Table S1: title, Video S1: title.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, J. C. Juang and C. M. Liu; methodology, J. C. Juang and C. M. Liu; software, C. M. Liu; data

curation, C. M. Liu; writing—original draft preparation, C. M. Liu; writing—review and editing, J. C. Juang; supervision, J. C. Juang. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Acknowledgments: This paper is supported by the Ministry of Science and Technology (MOST), Taiwan, under grant MOST 109-2218-E-006-032.

References

1. Li, B.; Zhang, T.; Xia, T., Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916* **2016**.
2. Tian, B.; Yao, Q.; Gu, Y.; Wang, K.; Li, Y. In *Video processing techniques for traffic flow monitoring: A survey*, 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011; IEEE: 2011; pp 1103-1108.
3. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.-Y., Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems* **2015,** 16, (2), 865-873.
4. Polson, N. G.; Sokolov, V. O., Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies* **2017,** 79, 1-17.
5. McCulloch, W. S.; Pitts, W., A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **1943,** 5, (4), 115-133.
6. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G., Human-level control through deep reinforcement learning. *Nature* **2015,** 518, (7540), 529.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. In *You only look once: Unified, real-time object detection*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; 2016; pp 779-788.
8. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M., Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* **2015,** 115, (3), 211-252.
9. Redmon, J.; Farhadi, A., Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**.
10. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y. M., Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**.
11. Wang, C.-Y.; Liao, H.-Y. M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. In *CSPNet: A new backbone that can enhance learning capability of CNN*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020; 2020; pp 390-391.
12. Ciaparrone, G.; Sánchez, F. L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F., Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020,** 381, 61-88.
13. Wojke, N.; Bewley, A.; Paulus, D. In *Simple online and realtime tracking with a deep association metric*, 2017 IEEE international conference on image processing (ICIP), 2017; IEEE: 2017; pp 3645-3649.
14. Wang, Z.; Zheng, L.; Liu, Y.; Wang, S., Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605* **2019,** 2, (3), 4.
15. Fedorov, A.; Nikolskaia, K.; Ivanov, S.; Shepelev, V.; Minbaleev, A., Traffic flow estimation with data from a video surveillance camera. *Journal of Big Data* **2019,** 6, (1), 1-15.
16. Santos, A. M.; Bastos-Filho, C. J.; Maciel, A. M.; Lima, E. In *Counting Vehicle with High-Precision in Brazilian Roads Using YOLOv3 and Deep SORT*, 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2020; IEEE: 2020; pp 69-76.
17. Punn, N. S.; Sonbhadra, S. K.; Agarwal, S., Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. *arXiv preprint arXiv:2005.01385* **2020**.

18. Qiu, Z.; Zhao, N.; Zhou, L.; Wang, M.; Yang, L.; Fang, H.; He, Y.; Liu, Y., Vision-Based Moving Obstacle Detection and Tracking in Paddy Field Using Improved Yolov3 and Deep SORT. *Sensors* **2020**.
19. Seibold, B., A mathematical introduction to traffic flow theory. Mathematical Approaches for Traffic Flow Management Tutorials. In Institute for Pure and Applied Mathematics, UCLA.
20. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. In *Microsoft coco: Common objects in context*, European conference on computer vision, 2014; Springer: 2014; pp 740-755.