Low Computational Vehicle Lane Changing Prediction Using Drone Traffic Dataset

Youlkyeong Lee, Qing Tang, Jehwan Choi, Kanghyun Jo
Dept. of Electrical, Electronic and Computer Engineering
University of Ulsan, Ulsan, Korea
{yklee, tangqing, jehwan}@islab.ulsan.ac.kr, acejo@ulsan.ac.kr

***Abstract*—Safe autonomous driving assistance systems are actively being developed based on Convolutional Neural Network (CNN). Unlike understanding the road environment through the image viewed from the existing vehicle, it has the advantage of a drone image that can see a large area at once. It is used as safe driving assistance information by understanding the movements of various vehicles and predicting movement information according to time. In this paper, vehicle movement is predicted using LSTM by extracting vehicle time series information. Use YOLOv5 to detect the vehicle on the road. Road areas are collected as drone flight images. YOLOv5 is learned by labeling the vehicle through the collected image. Time-series vehicle movement information is extracted from the detected vehicle and the movement of each vehicle is predicted using the LSTM model. The predicted vehicle information is represented by an error through the MSE.**

***Index Terms*—Drone flight image, Long-short term memory, object detection**

2022 International Workshop on Intelligent Systems (IWIS) | 979-8-3503-9893-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWIS56333.2022.9920801

I. Introduction

Recently, smart mobility has been developed and used in real life. Self-driving vehicles are usually the most familiar. Various studies such as object classification [1], [2], object detection [3]–[5], and object re-identification [6]–[8], and traffic environment analysis are being actively conducted through computer vision algorithms using multiple cameras in vehicles. Additionally, the usability of drones, which are emerging as new mobility, is endless. In particular, drones can comprehensively monitor large areas different from vehicles, and there are no restrictions on movement, a various kinds of application developments are carried out by processing abundant information such as intelligent traffic information analysis, object tracking, and monitoring systems. However, there is still a lack of drone detection data sets required for traffic analysis, making it difficult to analyze general traffic information. It is expected that next-generation smart autonomous vehicles adopt the advanced driver assistance system, ADAS, and self-driving system based on high safety by utilizing not only information through vehicles, but also sensor and image information collected by drones. In Fig 1, the traffic drone image collected primarily generates motion information for an individual vehicle. Next, the condition of the vehicle on the road is analyzed through either the Convolutional Neural Network or the Recurrent Neural Network.

Currently, intelligence systems through various sensors and cameras in vehicles are being introduced. When pedestrians and drivers fail to monitor the danger around them at all times, dangerous situations often occur because it is difficult to make instant judgments in a short time. Future mobility not only detects risks through high-level intelligent surveillance systems but also helps predict expected situations through existing information. Drones, which can grasp large areas at a glance, predict future situations by utilizing recurrent neural networks (RNNs) through continuous images. The predicted information will be used as important information for the operation of ADAS and self-driving systems, which are safe for intelligent mobility systems.

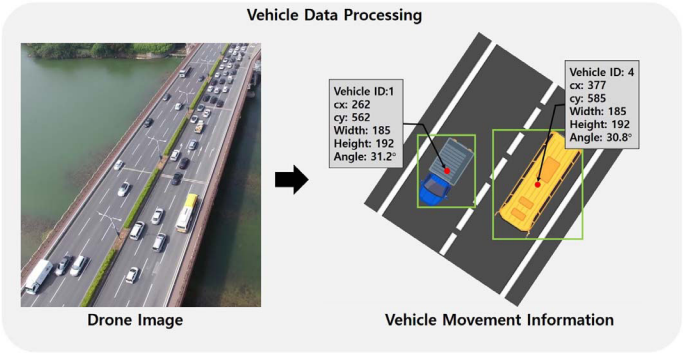


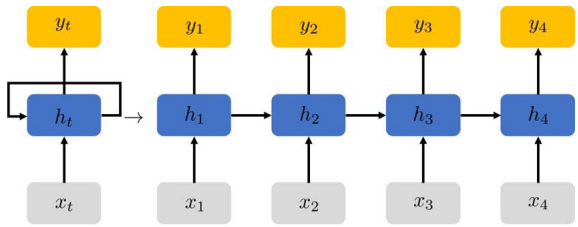
Fig. 1: The illustration of vehicle data processing for extracting vehicle information.

This study conducts 1) vehicle detection, 2) vehicle re-identification, 3) vehicle motion data collection, 4) Sequential Data training model development, and 5) vehicle location prediction through drone flight images. The following session presents the contents of previous studies of the proposed method.

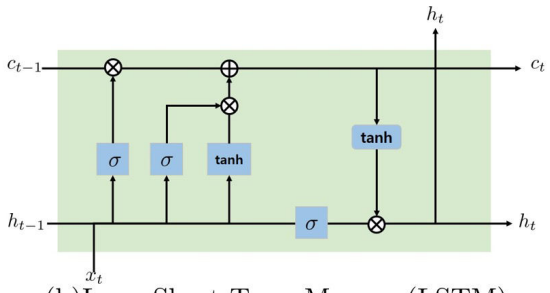
II. Related work

A. Recurrent Neural Network

In Fig. 2(a), Recurrent Neural Network (RNN) is a model that processes input and output values in sequence to analyze a series of data. The input value x_t stores information summarized in the hidden state. The sequential data updates the summarized input information through the hidden state. Therefore, this work develops a sequence model by iterating the time series data cyclically. However, if the circulation method is simply continued, the concentration of the previ-



(a)Recurrent Neural Network(RNN)



(b)Long Short Term Memory(LSTM)

Fig. 2: (a)Structure of Recurrent Neural Network; (b)Process of Long Short-Term Memory

ously remembered information will fade and the performance of the initial information will be lost. In 1997, Long Short-term Memory (LSTM) [9] overcomes the short-term process with the sequential data. In Fig. 2(b), LSTM contains a hidden state, h_t for the short-term state, and a cell state, c_t for the long-term state. There are several steps for updating the next cell, c_t and hidden, h_t state. First, the cell state keeps the information from the previous state, c_{t-1}. The process adopts the linear interaction to store the previous information. Second, the gate which carefully filters the important data via input and h_{t-1}, controls to modify of the cell state. There are forget gate layer, an input gate layer, and an output gate layer. With these layers, x_t and h_{t-1} selects the meaningful information to update c_t and h_t. Finally, both a new cell and hidden recurrently run the next sequential data.

B. Object Re-Identification

This study is followed by data extracted after re-identification processing of vehicles in continuous images. Object re-identification is an area that requires inter-frame data analysis based on consecutive image processing such as video images. First, it is necessary to detect objects in the image. In the image, object detection is a two-stage object detection algorithm that sequentially checks region proposal and classification including Fast R-CNN [3] and Mask R-CNN [4]. In addition, one-stage object detection, YOLO [5], [10]–[12] and Focal Loss [13], which simultaneously processed region proposals and classification, are mainly used. Object re-identification compares the characteristics of objects between consecutive frames and gives the same object ID. The conventional approach begins with re-identification for people. Characteristic information on a person's shape, color, and texture are extracted to determine the similarity with a person

Fig. 3: Within cropped vehicle area to extract edge; first row: cropped image in the original, second row: edge image from a cropped image.

detected by a camera in another frame or another location to have a person's ID. In this way, a similar method is used for the re-identification method applied to the vehicle. VOC-ReID, [14], the triplet vehicle-orientation-camera considers the background and shape similarity as camera and orientation re-identification. Strong Baseline, [15] contains domain generalization, attention mechanism and ID/Metric loss function for vehicle re-identification. FairMOT, [8] simultaneously handles detection and re-identification. At the same time, the process considers balanced high detection performance and tracking accuracy. This paper, this paper conducts a study to predict vehicle location using the LSTM model that analyzes vehicle movement information and sequential vehicle movement information organized through vehicle re-identification.

III. Proposed Method

A. Sequential Traffic Data

With the drone dataset, the detector is trained for detecting the vehicle. In this study, sequential traffic data are used as input values for RNN models. First, the vehicle is detected through YOLOv5 [12] algorithm. The information of the detected vehicle on the road is as follows:

- (x,y), coordinate
- Vehicle of width and height

The Bounding box information of the vehicle detected through the YOLOv5 [12] model generates center coordinates (cx, cy) and sizes (width, height). Additionally, the vehicle position of angle shows one of status for vehicle movement. The inclined angle of the vehicle in the detected area extracts an edge through the Sobel filter [16]. In Fig. 3, it shows cropped region for extracting edge using Sobel filter. The edge is extracted for the x and y directions, and radians are calculated through radian = arctan (Gy/Gx). Radian is converted to degree and designates the average of the angle values generated in the area as the tilted angle of the vehicle. Angle represents average angle for each vehicle where i an j are the pixel position in the cropped area. The Eq.(1) denotes as follows:

$$\text{Average angle} = \frac{1}{w * h} \sum_{i=1}^{w} \sum_{j=1}^{h} \mathbb{A}_{ij} \tag{1}$$

As a sequential vehicle movement data, vehicle re-identification needs to identify each vehicle to collect the information for every frame. Traditionally, for re-identification,

features for the corresponding object are compared and identified between frames are identified. The feature extraction method is applied to the detected vehicle area, but in this study, vehicle re-identification is carried out by comparing the distance between frames for the detected vehicle. The similarity of vehicle distances between frames is calculated as Euclidean distances. Moving distance measures the moving distance as a pixel-level between the detected vehicle $V_{k,i}$, in i-th frame and $V_{k,i+1}$, in $i+1$-th frame. k is the index of the detected vehicle. Eq.(2) calculates the moving distance of the vehicle both current and next frames. $V_{k,i} = (x_{k,i}, y_{k,i})$ is the center coordinates of $V_{k,i}$. Therefore, the moving distance between $V_{k,i}$ and $V_{k,i+1}$ as follows:

$$D(V_{k,i}, V_{k,i+1}) = \sqrt{(x_{k,i} - x_{k,i+1})^2 + (y_{k,i} - y_{k,i+1})^2} \quad (2)$$

The movement data of the sequential vehicle is defined as shown in Table 1. Sequential vehicle movement data is a set

TABLE 1: Illustration for each vehicle of sequential data

Frame	cx	cy	Width	Height	Angle
frame1	**1566**	**2062**	**172**	**182**	**30.34**
frame2	**1577**	**2041**	**172**	**182**	**30.28**
frame3	**1588**	**2021**	**170**	**184**	**30.92**
⋮	⋮	⋮	⋮	⋮	⋮

of frame $\mathbb{R}^{n\times 5}$ that collects [cx, cy, Width, Height, Angle] for each vehicle. The batch size contains a series of the frame to understand the consecutive data. When this work selects and learns the vehicle moving information it is limited to at least 50 frames of vehicle. If the set of the frame is smaller than 50, it is difficult to figure the lane changing the vehicle out.

B. Recurrent Neural Network for Sequential Data

In order to predict the movement of the vehicle, this study attempts to predict the movement of the vehicle through the recurrent neural network model with sequential input data. The input value is motion information of the vehicle, (cx, cy, width, height, angle). RNN has a hidden layer and updates the previous hidden layer to remember past weight information and reflect it in the learning. However, as time goes by, RNN forgets the weight memory information from the initial time. This is a vanishing gradient problem. Short-term memory can produce meaningful results, but it cannot leave long-term meaning behind. Long Short Term Memory (LSTM) [9] algorithm overcomes the problem. The operation inside the network solves the vanishing gradient problem by adding a plus operation. In Eq.(3), the equation is related to the method of the LSTM as follows:

$$\begin{aligned} f_t &= \sigma(W_{xh_t} x_t + W_{hh_f} h_{t-1} + b_{h_f}) \\ i_t &= \sigma(W_{xh_i} x_t + W_{hh_i} h_{t-1} + b_{h_i}) \\ o_t &= \sigma(W_{xh_o} x_t + W_{hh_o} h_{t-1} + b_{h_o}) \\ g_t &= \tanh(W_{xh_g} x_t + W_{hh_g} h_{t-1} + b_{h_g}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (3)$$

The forget gate, f_t is a gate for forgetting past information. σ is a sigmoid function, and the value obtained by calculating h_{t-1} and x_t is f_t. The sigmoid expresses a value between 0 and 1 and forgets the previous information if the value is 0, and fully remembers the previous state if it is 1. Input gate, $i_t \odot g_t$ is a current memory information. i_t is in the range (0,1) and g_t is in the range (-1,1). It shows strength and direction, respectively. The LSTM is trained with Sequential data to analyze the trend of sequential data to generate predictions for future vehicle movements.

IV. EXPERIMENT

The drone dataset was created looking from the drone toward the road. The resolution of image is 3840×2160(4K). There are 9,776 images used for training for object detection and 2,200 images used for testing. The entire object in the image used for the training is 309,470 cars and trucks annotated.

Implementation Details: For training on object detection, existing images were learned by reducing them to 960x960. The batch size is 16. The learning rate is 0.01 and uses OneCycleLR [17] for learning schedule. Pytorch [18] is a main framework. The Graphic card uses RTX 3090 GPU and 32G RAM memory. The evaluation metrics comes from MS-COCO [19]. The mean average precision AP_{50} and $AP_{50:95}$ are usually taken to measure object detection performance.

Object Detection: To analyze vehicle movement on the road, YOLOv5 [12] is first adopted to proceed with object detection. Cars and trucks are the main objects for sequential image data. Table 2 represents the detection performance. Training performance is 95.75% for mAP@50 and 83.8% for mAP@50:95. The test performance is 91.8% for mAP@50 and 80.3% for mAP@50:95. Input data of the LSTM model is generated through a result of detecting a vehicle of all frames through the corresponding learning model. As shown in Fig. 4,

TABLE 2: Result of object detection with drone dataset

Class	Images	Instance	mAP@50	mAP@50:95
all_train	**9,776**	**309,470**	**95.75**	**83.8**
car_vehicle	**9,776**	**277,263**	**97.2**	**86.0**
truck_vehicle	**9,776**	**32,207**	**94.3**	**81.6**
all_test	**2,200**	**85,398**	**91.8**	**80.3**
car_vehicle	**2,200**	**78,765**	**96.1**	**85.3**
truck_vehicle	**2,200**	**6,633**	**87.5**	**75.4**

the red and blue lines are the position values of the vehicles predicted through the trained model, and the red and blue dash lines are the ground truth values. Fig. 4(a) is a result of vehicle ID: 9, and Fig. 4(b) is a result of vehicle ID: 14. It can be expressed as if the error between the predicted value and the actual value is large, but it can be seen that the trend for the vehicle's movement line results in a similar result to the actual value. Table 3 shows the difference between ground truth and prediction. The predicted result value for all frames of each CarID is a Mean Square Error (MSE) calculation value with ground truth. The closer the error value is to 0, the closer it is to a suitable model.

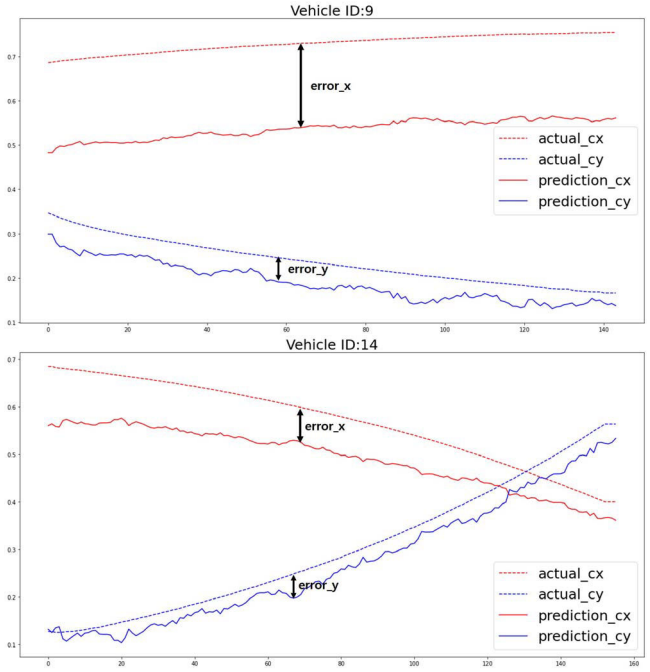


Fig. 4: Illustration for predicting vehicle position with ground truth data, (a) shows vehicle ID:9, (b) shows vehicle ID: 14, red and blue dash lines are ground truth of center x and y coordinate, red and blue lines are predicting of center x and y coordinate.

TABLE 3: Result of error x and y coordinate

CarID	9	14	22	32	38	46
error_x	0.0373	0.0064	0.0049	0.0298	0.0121	0.0059
error_y	0.0023	0.0010	0.0139	0.0549	0.0025	0.0018
CarID	57	68	80	85	94	101
error_x	0.0047	0.0041	0.0033	0.0023	0.0060	0.0078
error_y	0.0015	0.0061	0.0077	0.0040	0.0024	0.0003

V. Conclusion

This study proposes a method of predicting the movement of a vehicle by analyzing the sequential data generated using LSTM. Unlike image data collected through a conventional vehicle, it is a method of predicting the movement of a vehicle predicted by an image captured by a drone. The object detector applies YOLOv5. In the future, the next work will improve the prediction model by upgrading the time series analysis model, and developing it so that real-time analysis is possible on the edge device, NVIDIA jetson nano, or Raspberry pi.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[2] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[3] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020.

[5] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

[6] Chao Liang, Zhipeng Zhang, Yi Lu, Xue Zhou, Bing Li, Xiyong Ye, and Jianxiao Zou. Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, 31:3182–3196, 2022.

[7] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *ArXiv*, abs/1909.12605, 2020.

[8] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129:3069–3087, 2021.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[10] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[11] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[12] Glenn R. Jocher, Alex Stoken, Jiří Borovec, NanoCode, Ayushi Chaurasia, TaoXie, Liu Changyu, Abhiram, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang, Jan Hájek, Laurentiu Diaconu, Marc, Yonghye Kwon, Oleg, wanghaoyang, Yann Defretin, Aditya Lohia, ml ah, Ben Milanko, Ben Fineran, D. P. Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - yolov5-p6 1280 models, aws, supervise.ly and youtube integrations. 2021.

[13] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2020.

[14] Xiangyu Zhu, Zhenbo Luo, Pei Fu, and Xiang Ji. Voc-reld: Vehicle re-identification based on vehicle-orientation-camera. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2566–2573, 2020.

[15] Su V. Huynh, Nam-Hoang Nguyen, Ngoc-Thanh Nguyen, Vinh Nguyen, Chau Huynh, and Chuong H. Nguyen. A strong baseline for vehicle re-identification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4142–4149, 2021.

[16] Irwin Sobel and G. M. Feldman. An isotropic 3×3 image gradient operator. 1990.

[17] Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Defense + Commercial Sensing*, 2019.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.