# A Neural Network-based Multisensor Data Fusion Approach for Enabling Situational Awareness of Vehicles

Albert Budi Christian†, Chih-Yu Lin*, Cheng-Wei Lee†, Lan-Da Van†, and Yu-Chee Tseng†
* Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan
Email: lincyu@mail.ntou.edu.tw
† Department of Computer Science, National Chiao Tung University, Taiwan
Email: albertchrist92.eed06g@nctu.edu.tw, davidlee831020@gmail.com, ldvan@cs.nctu.edu.tw, yctseng@cs.nctu.edu.tw

***Abstract*—With the growing number of research studies on Vehicle-to-Vehicle (V2V) communication applications, situational awareness becomes one of major challenges for autonomous vehicles. Autonomous vehicle needs to predict the movement and trajectories of surrounding vehicles accurately in order to make a better decision making. The ability to recognize vehicles' surroundings has become important in order to enable situational awareness and navigate the vehicle safely. In this paper, we propose a neural network called Mapping Decision Feedback Neural Network (MDFNN) to tackle the vehicle identification (VID) issue in V2V communication. According to the MDFNN infrastructure, two types of MDFNN namely as Grid-based MDFNN and Bounding box-based MDFNN are proposed. The MDFNN fuses image, V2V interface, GPS, magnetometer, and speedometer data (i.e., multi-sensor data and V2V communication) to enable situational awareness. MDFNN utilizes the mapping decision feedback information in the proposed deep learning neural network structure. With this improvement, a greatly improved accuracy can help to resolve the VID issue. Our experiment's result shows 85% of accuracy for Grid-based MDFNN.**

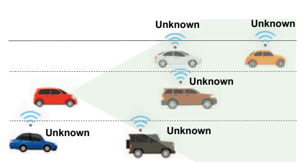
***Index Terms*—automomous driving, vehicle identification (VID), data fusion, deep learning, neural network, V2V communication**

## I. Introduction

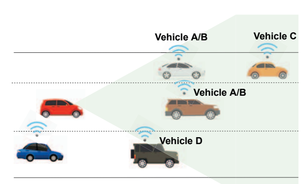
Recently, utilizing numerous sensors to provide a better information about vehicle's surrounding on autonomous vehicles has become an interesting research topic. The ability to understand the behavior of surrounding vehicles can be referred as situational awareness [1]. To enable situational awareness, an accurate vehicle's status such as location and speed must be able to be delivered properly [2]. While many researches targeting on vehicle-to-vehicle (V2V) communication, it does not imply that the vehicle can fully depend on this particular system to have full perception of its surroundings [3] [4].

Data transmission between vehicles must be strengthen and well integrated to make it more meaningful with the purpose of making a better decision for autonomous driving vehicle. On the other hand, although the idea of deep learning has been appeared earlier than 1980 [5] [6], it has just become popular at the past decades to provide a better decision maker for autonomous driving vehicle. With the rapid advancing technology development, Convolution Neural Network (CNN) has been proved to be a powerful deep learning models, therefore can help provide a more accurate decision-maker on autonomous driving vehicle [7].

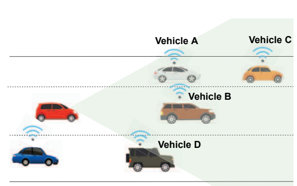
In this paper, our goal is to achieve a better situational awareness by fusing multi-sensor data and benefiting from V2V communication which allows information exchange between vehicles. However, a challenge occur on how to identify which vehicle whose send the information. Therefore, in this paper, we propose a neural network called Mapping Decision Feedback Neural Network (MDFNN) infrastructure to tackle VID issue. An example is shown in Fig. 1. In Fig.1a, there are five vehicles surrounding the red vehicle. The red vehicle receives broadcasts from those five vehicles via V2V communications. However, a dash camera that has been installed in the red vehicle can only detects four vehicles ahead as shown in Fig. 1b. The goal of this paper is to correctly identify each broadcast message's sender to its corresponding vehicle shown by the image in 1c. In the end, it will be visualized in V2V interface as shown in Fig. 1d.




(a) A road scenario.




(b) The result of vehicle locator.




(c) Final result of MDFNN.


(d) Visualization.

Fig. 1. Vehicle identification (VID) problem

978-0-7381-4262-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ICPAI51961.2020.00044

The main contributions of this paper are described as follows.

1) We propose a new deep learning neural network called Mapping Decision Feedback Neural Network (MDFNN) infrastructure that considers vehicles' information and V2V communication information and past decision mapping results as the input to tackle VID issue. To our best knowledge, compared with other neural networks like recurrent neural network (RNN) [8], the proposed MDFNN infrastructure utilizes the previous iterative mapping decision information in order to improve the VID accuracy.
2) Based on MDFNN infrastructure, we propose two type neural networks: Grid-based MDFNN and Bounding box-based MDFNN. The Grid-based MDFNN divides the image into several grids and estimates the probability of which grid(s) covers the target surrounding vehicles. Through the Grid-based MDFNN, the VID accuracy can be improved by 35% compared with our baseline neural network. On the other hand, the Bounding box-based MDFNN estimates the position of the target surrounding vehicles. Utilizing the Bounding box-based MDFNN, the VID accuracy can be improved by 30% compared with our baseline neural network. Accordingly, two loss functions are provided.
3) The corresponding mapping decision score functions are presented to evaluate the similarity of an estimated vehicle position and a real vehicle position on the image.

The rest of this paper is organized as follows. Section II reviews some related works. Section III introduces our system model. Section IV shows the experiment results. Conclusion are drawn in Section V.

## II. Related Work

To prevent accident and make better decisions for autonomous vehicles, improving the accuracy of situational awareness is important. While a radar technology has been around and has a great value for obtaining distance and speed [9], driving status is relatively hard to be collected from this technology [10]. On the other hand, camera has gain popularity to be utilized to acquire a situational awareness. Benefiting from dash camera, [11] has already proposed a system for distance estimation.

An improvement to measure the speed of the vehicle has also been proposed in [12]. In [13], a monocular camera is used for identifying and tracking preceding vehicles. However, due to heavy traffic with many vehicles coming in a complex direction, the accuracy on vehicle detection is not enough efficient. V2V communication can be used for improving road safety and efficiency [14]. In [15] has proposed a collision warning system by using V2V technology. However, this proposed solution still has a room to improve regarding the information based on vehicle's surrounding. The required data rate on V2V is derived in [16] to ensure safe autonomous driving in an overtaking driving situation.

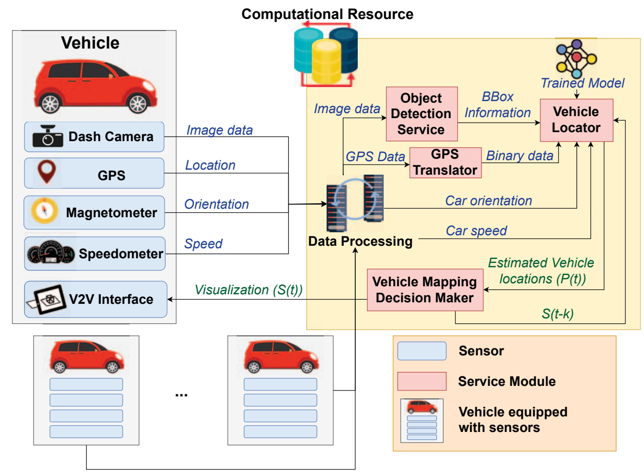



Fig. 2. System model of MDFNN.

Multiple technologies can be integrated to enable situational awareness. For example, [17] has proposed a vehicle following control system by integrating radar and V2V communication. Nevertheless, to obtain an effective accuracy in the communication system using spectrum sharing, a robust interference should be suppressed by a great amount. This solution relies on two stereo cameras equipped on vehicle. Thus, it can easily blocked by bus or truck. The data fusion technique improves localization accuracy by adopting multiple sensory data of mapping. In [18], a data fusion approach has been proposed to match the image data captured by the camera and the radio data received via V2V communication. In this paper, we propose a deep learning neural network to tackle the VID problem as detailed in Section III.

## III. System Model

The system model of MDFNN is shown in Fig. 2. Each vehicle is equipped with a dash camera to continuously records the view through a vehicle's front windscreen. It also embed an on-board units (OBU) which contains GPS, V2V interface, magnetometer, and speedometer. Each vehicle will periodically broadcast a message containing its profile and driving status to surrounding vehicles. The broadcast message contains following information:

- Vehicle profile: plate number and GPS information.
- Driving status: current speed, breaking status, and turning status.

To prevent privacy concern, a vehicle may broadcast its $H(pn)$, where $pn$ is its plate number and $H()$ is a hash function. Therefore, the plate number can only be used to identify whether two broadcasts are sent from the same vehicle. In this case, we can consider $pn = pn'$ if $H(pn) = H(pn')$.

We define the VID problem as follows. Assuming a vehicle is denoted by $x$. At each time $t$, $x$ retrieves its location $x(t).loc$, speed $x(t).sp$, and orientation $x(t).ot$ from its GPS, magnetometer and speedometer. In this case, we use discrete time in our representation, so $t$ means the $t-th$ time

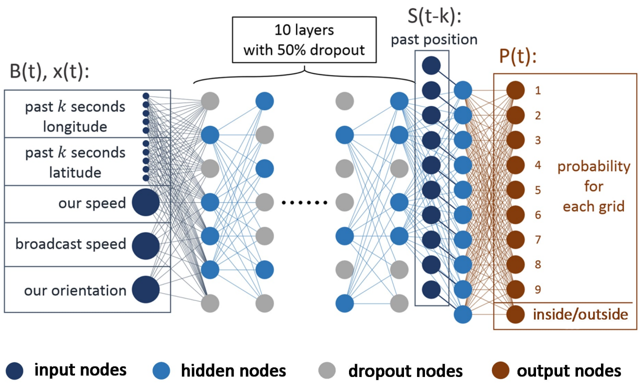



Fig. 3. The neural network structure of the grid based MDFNN.

unit. $x$ takes an image $I(t)$ from camera and obtains $n$ vehicles' bounding boxes $V(t) = \{v_i(t)|1 \leq i \leq n\}$. $x$ also receives $m$ broadcasts $B(t) = \{b_j(t)|1 \leq j \leq m\}$ by V2V communication where each $b_j(t)$ contains location $b_j(t).loc$, speed $b_j(t).sp$, and plate number $b_j(t).pn$. Then our goal is to find the correct matching pairs $S(t)$, where:

$$S(t) = \{(v_i(t), b_j(t))|v_i(t) \in V(t) \text{ and } b_j(t) \in B(t)\} \quad (1)$$

A fusion between images and broadcasts data is needed in order to recognize vehicles surrounding. This ability proven useful to enable situational awareness in order to navigate the vehicle safely. Each module in Fig. 2 is demonstrated as follows:

### A. *Object Detection Service*

The images captured by dash camera are the input of object detection service module, which will generate $n$ vehicles' bounding boxes denoted by $V(t)$. To identify the vehicles, we use YOLO [19] to find the bounding boxes of vehicles on the image. The bounding box is stored as the coordinate of its top left corner and bottom right corner, with the format: $v_i = [(x_{TopLeft}, y_{TopLeft}), (x_{BottomRight}, y_{BottomRight})]$, where $v_i$ means the bounding box of the $i-th$ vehicle in the image $I$.

### B. *GPS Translator*

In this module, the broadcast data and $x$'s sensor data is processed into the form we preferred. We calculate the difference between GPS location obtained from the received broadcast and $x$ itself. For example, if a broadcast message indicates a broadcast data at (2358.3327 N, 12058.26102 E) and $x$ is located at (2358.3234 N, 12058.25975 E), then the difference is (0.0093, 0.00127). Finally, the GPS translator module normalize all data's value to be between 0 and 1 (both inclusive).

### C. *Vehicle Locator (MDFNN)*

In vehicle locator module, we propose the MDFNN infrastructure to estimate the position of the broadcast-sender vehicles on the image. This module is typically inspired by Recurrent Neural Network (RNN) [8]. While RNN directly uses its result as an input, our proposed neural networks has a main feature which utilize the previous iterative result for its input.

We present two types of MDFNN on this module: Grid-based MDFNN and Bounding box-based MDFNN. In Grid-based method, we divide the image into many grids. Furthermore, the output of the neural network is the probabilities of grids where the vehicle is detected. On the other hand, for the Bounding box-based MDFNN, the output is the bounding box's itself.

*1) Grid-based MDFNN:* The neural network of the Grid-based MDFNN is shown in Fig. 3. The input of neural network includes: the longitude difference from $k$ seconds ago until now, the latitude difference from $k$ seconds ago until now, the speed of our vehicle $x$, the speed of broadcast-sender vehicle obtained from the broadcast message, the orientation of $x$, and the $k$ seconds ago position of the broadcast-sender vehicle. There are 10 middle layers with 50% dropout and a fully connected layer before the output layer. The output will be an array of $n+1$ instances where $n$ is the number of the grids (e.g. if we partition the image into $5*5$ grids, then the output will be an array with 26 instances).

To label a broadcast sent by broadcast-sender vehicle $y$, we do the following:

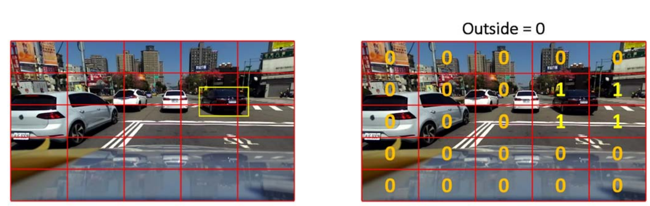
- In the previous $n$ instances if vehicle $y$ is not in the $i-th$ grid, the value of the $i-th$ instance = 0.
- If the intersection area of the $y$'s bounding box and grid is greater than a threshold (threshold = 10%), the value of the $i-th$ instance = 1.
- If vehicle $y$ is on the image, the value of the last instance = 0.
- If vehicle $y$ is not exist on the image, the value of the last instance = 1.

An example is shown in Fig. 4, where we assume that received broadcast is sent by the black vehicle on the image and we divide the image into $5*5$ grids. Then the broadcast is labeled as shown on the right hand side of Fig. 4.

The loss function of the Grid-based MDFNN is defined as follows:

$$Loss_g = -\frac{1}{m}\Sigma_{j=1}^{m}\frac{\Sigma_{i=1}^{n}\log(y_{ji}p_{ji})}{\Sigma_{i=1}^{n}y_{ji} + y_j^o} + \alpha\frac{-\Sigma_{j=1}^{m}\log|y_j^o - p_j^o|}{m} \quad (2)$$

The first part of eq. 2 indicates the position of a vehicle on the image. The second part of eq. 2 is defining that a broadcast is exist on the image, where $m$ is the number of data, $n$ is the




(a) Initial state: Grid without label. (b) Final state: Grid with label.

Fig. 4. An example of the Grid-based MDFNN labeling.

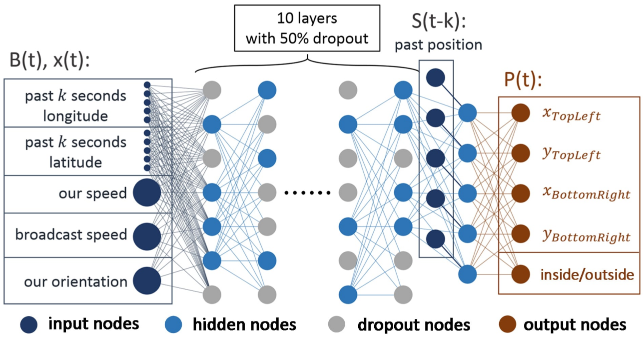



Fig. 5. The neural network structure of the Bounding box-based MDFNN.

number of grids, $y_{ji}$ is the ground truth of the broadcast's position. Furthermore, $p_{ji}$ is the output of the broadcast's position, while $y_j^o$ is whether the ground truth of the broadcast is inside the image. $p_j^o$ is define as the probability output of the broadcast on the image, and $\alpha$ is the importance of the second part of the loss function.

*2) Bounding box-based MDFNN:* The neural network of the Bounding box-based MDFNN is shown in Fig. 5. The input layer and the middle layers are similar to the neural network of the Grid-based MDFNN, while the output of this process has five instances: previous four instances are the bounding box and the last instances is the availability of the vehicle on the image. Fig. 6 is an example for labeling a broadcast for the Bounding box-based MDFNN. The label of bounding box is transformed from $[(x_{TopLeft}, y_{TopLeft}), (x_{BottomRight}, y_{BottomRight})]$ to $[(x_{TopLeft}/w, y_{TopLeft}/h),$ $(x_{BottomRight}/w, y_{BottomRight}/h)]$ where $w$ and $h$ are the width and height of the image, respectively.

The loss function of the Bounding box-based MDFNN is defined as follows:

$$Loss_{bb} = \frac{\Sigma_{j=1}^{m}\Sigma_{i=1}^{4}(y_{ji}-p_{ji})^2}{m} + \alpha\frac{-\Sigma_{j=1}^{m}\log|y_j^o - p_j^o|}{m} \tag{3}$$

The first part is the loss for the bounding box and the second part is the loss for whether the broadcast is in the image, where $m$ is the number of data, $y_{ij}$ is the ground truth of the broadcast's bounding box, $p_{ji}$ is the output of the broadcast's bounding box, $y_j^o$ is the ground truth of the available broadcast inside the image, $p_j^o$ is the output of the available broadcast inside the image, and $\alpha$ is the importance of the second part of the loss function.

TABLE I
AN EXAMPLE OF SCORE TABLE

| Score | $v_1$ | $v_2$ | $v_3$ |
|---|---|---|---|
| $p_1$ | 0.85 | 0.88 | 0.5 |
| $p_2$ | 0.1 | 0.7 | 0.2 |

## *D. Vehicle Mapping Decision Maker*

After performing the vehicle locator module, we obtain an estimated position for each broadcast. Since there might be more than one vehicle in the image and/or more than one broadcast, therefore the next step is to match the corresponding pair together. To determine the similarity of an estimated position and a bounding box on the image, we define a function to calculate the score of similarity for each network of Vehicle Locator. The higher the score is, the higher chance that estimated position is corresponding to that bounding box.

For the Grid-based MDFNN, the score function is defined as follows:

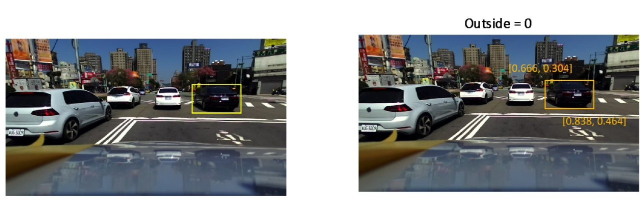
$$Score_g(p_j, v_l) = \frac{\Sigma_{i=1}^{n} p_{ji} * v_{li}}{\Sigma_{i=1}^{n} v_{li} + v_l^o} \tag{4}$$

where $p_j$ is an estimated position and $v_l$ is a bounding box on the image. In Fig. 7, the upper left image is the label of the white vehicle next to the black vehicle, the lower left image is the label of the black vehicle, and the right image is an estimated position of a broadcast. The score of the white vehicle and the estimated position is $(0.52 + 0.89 + 0.48 + 0.78)/4 = 0.67$ and the score of the black vehicle and the estimated position is $(0.89 + 0.72 + 0.78 + 0.85)/4 = 0.81$. Ultimately the broadcast is corresponding to the black vehicle.

For the Bounding box-based MDFNN, the score function is defined as the intersection over union (IoU):

$$Score_{bb}(p_j, v_l) = \frac{Area\ of\ Intersection(p_j, v_l)}{Area\ of\ Union(p_j, v_l)} \tag{5}$$

Taking from the same image as Fig. 7, by utilizing the Bounding box-based MDFNN as shown in Fig. 8, it shows




(a) Initial state: Bounding box without label.

(b) Final state: Bounding box with label.

Fig. 6. An example of the Bounding box-based MDFNN labeling.

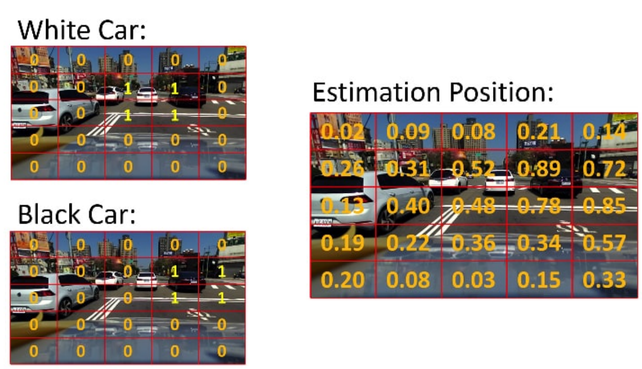



Fig. 7. An example of calculating score of the Grid-based MDFNN.


Fig. 8. An example of calculating score of the Bounding box-based MDFNN.

that the score for the white vehicle is 0.85, while the score of black vehicle is 0.88. Therefore these similar score can be assembled as shown in Table I.

The most straightforward method to select a corresponding pair is to select the highest score. In Table I, although the highest score is $p_1$ and $v_2$, the scores of $p_1$ and others are also high. On the other hand, the score of $p_2$ and $v_2$ is far higher than $p_2$ and others. Thus $\{(p_1, v_1), (p_2, v_2)\}$ might be a better choice. To solve this situation, we define a confidence function to choose which $p_j$ should be select first:

$$Confidence(p_j) = \frac{\max_{l=1}^{n}\{Score(p_j, v_l)\}}{\Sigma_{l=1}^{n} Score(p_j, v_l)} \tag{6}$$

Intuitively, if the score of $p_j$ and $v_l$ is far higher than $p_j$ and others, then the confidence of $p_j$ will be higher.

In summary, the Vehicle Mapping Decision Maker module can be done by these following steps:

Step 1: Filter the instance for those estimated vehicle positions which available on the image (threshold = 0.5).

Step 2: Calculate the scores for estimated positions of the filtering process and get the score table.

Step 3: Calculate the confidence function for each estimated positions in the table.

Step 4: Analyzing the confidence score from the higher confidence to the highest score

Step 5: Match the broadcast message and the vehicle respectively as:

$$S(t) = \{(v_i(t), b_j(t)) | v_i(t) \in V(t) \text{ and } b_j t \in B(t)\} \tag{7}$$

## IV. Experimental Results

In this project, we utilize CARLA [20] simulator to design the experiment and evaluate the accuracy of the proposed Grid-based MDFNN and the Bounding box-based MDFNN.

### A. Simulation Environment

In this project, a deployed vehicle on the road called $x$ equipped with dash camera records the view through a vehicle's front windscreen periodically. Five more vehicles with diverse velocity have been deployed as well. Therefore, these five vehicles will not be able to move neither in front of nor behind $x$. As the image from the camera has been captured, orientation, speed and coordinate of each vehicles is recorded at the same time. The simulated environment has been arrange in order to allow the vehicles drive according to the road contour as shown in Fig. 9.

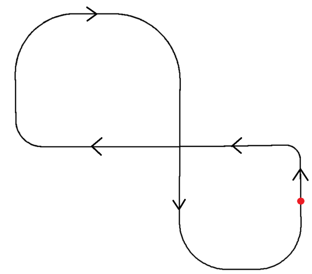


Fig. 9. The road contour in our experiment.

In order to simulate V2V communication, a broadcast messages are limited for vehicles in range of 30 meters from $x$. During these experiments, some situations are likely to happened as shown in Fig. 10. We utilize YOLO version 2 [21] to find the bounding boxes of each vehicle on the image. Note that the model provided by YOLO is trained by real vehicles. Since the vehicles in CARLA are a little different from real vehicles, there are about 5% of vehicle images cannot be detected by YOLO in the experiment.

The first situation as shown in Fig. 10a shows where other vehicles are located on one side of the dash camera's point of view. The second situation as shown in Fig. 10b shows other vehicles are spreading over the various side of dash camera's point of view. The third situation as shown in Fig. 10c is showing other vehicles are affected by the distance between the dash camera and its locations. While, the broadcast message still be received, however the other vehicles are not detected clearly by dash camera. All three

(a) Other vehicles are located on one side of dash camera.

(b) Other vehicles are located on various side of dash camera.

(c) Other vehicles are located on various side of dash camera.

(d) All vehicles stopped by traffic light.

Fig. 10. Probable situations during the experiment.

situations are affecting on vehicle identification system. Lastly, in the fourth situation as shown in Fig. 10d shows that all the vehicles are not broadcasting any vehicle status due to traffic light.

### B. Evaluation

To evaluate the MDFNN performance, we define the accuracy as following:

$$Accuracy = \frac{\Sigma_{\forall t}\ correct\ mapping\ broadcasts\ in\ t}{\Sigma_{\forall t}\ all\ broadcasts\ in\ t} \tag{8}$$

*1) Grid-based MDFNN:* Fig. 11 shows the accuracy with different numbers of grids. We verify the proposed system model through $5*5$ grids, $6*6$ grids, ..., and $12*12$ grids simultaneously. We compare the accuracy's result of using MDFNN, RNN, and baseline. In this case, baseline is MDFNN without mapping decision as input (i.e., no "past position" layer in Fig. 3). The result shows that the accuracy of MDFNN is about 85 % when the number of grids is more than $10*10$, while the accuracy of RNN is about 80% and baseline is about 50%.

In this work, the model has been tested by assuming the vehicle $x$ only receives one broadcast. Consequently, the Vehicle Mapping Decision Maker module are ultimately executed. There is a possibility that the final result of MDFNN are not sufficient. Therefore, the Vehicle Mapping Decision Maker module verify the mapping result. Since vehicle $x$ only receive one broadcast, the result of MDFNN will affect the final mapping result significantly which has been shown in Fig. 12. Although the accuracy of MDFNN is rather low as shown in Fig. 11 the RNN's result has been proven worse.

*2) Bounding box-based MDFNN:* The results of the Bounding box-based MDFNN is shown in Fig. 13. The VID accuracy of MDFNN is about 80% and RNN is about 77% which are both a little lower than the Grid-based MDFNN.

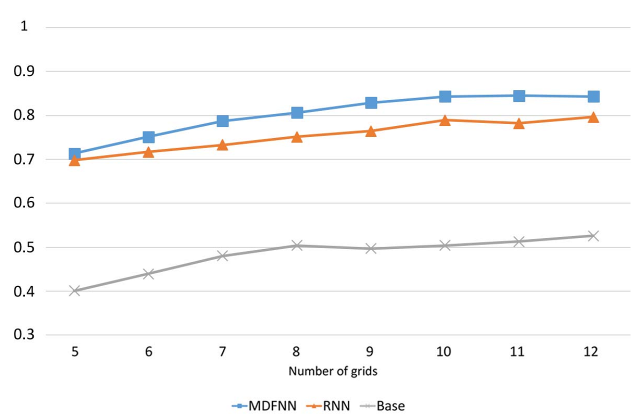



Fig. 11. VID accuracy of the Grid-based MDFNN.

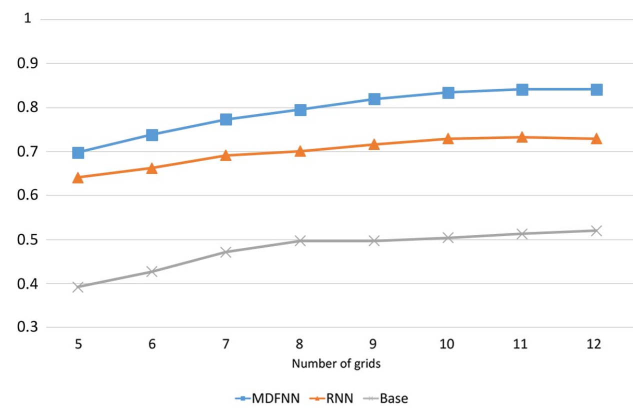



Fig. 12. Accuracy of the system which receiving one broadcast at a time.

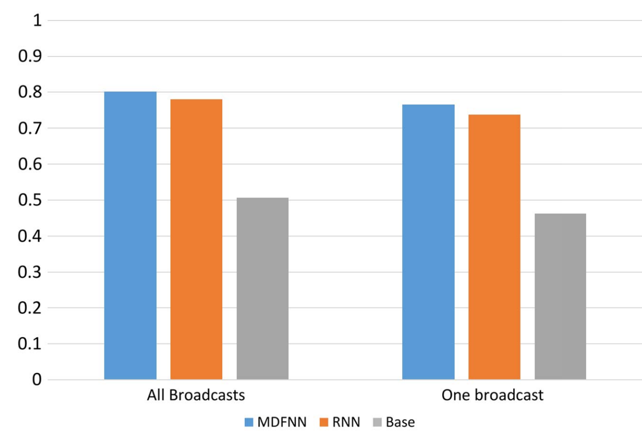



Fig. 13. VID accuracy of the Bounding box-based MDFNN.

## V. Conclusion

For autonomous vehicle, a situational awareness is vital to recognize vehicle's surrounding. Therefore, situational awareness is needed to provide a better decision making. To enable the situational awareness, MDFNN is proposed. This method allow us to estimate vehicle's position from a broadcast transmitted by surrounding vehicle via V2V communication using the images captured by dash camera. Although this proposed method is inspired by RNN, it has been proved that the result has a better accuracy. We proposed two types of neural networks namely Grid-based MDFNN and Bounding box-based MDFNN. While Bounding box-based MDFNN is simpler, the Grid-based MDFNN has a slightly higher VID accuracy. Two score functions and a confidence function are presented to identify which vehicle broadcast their status. Finally, we utilize CARLA to simulate on-road vehicles and compare the proposed two types of MDFNN, RNN, and baseline neural networks.

## Acknowledgement

Y.-C. Tseng's research is co-sponsored by ITRI, Pervasive Artificial Intelligence Research (PAIR) Labs, 105-2218-E-

009-003 (MoST), and ”Center for Open Intelligent Connectivity” (MoE).

## REFERENCES

[1] T. Fong and C. Thorpe, “Vehicle teleoperation interfaces,” *Autonomous robots*, vol. 11, no. 1, pp. 9–18, 2001.

[2] C. Wang *et al.*, “Designing for enhancing situational awareness of semi-autonomous driving vehicles,” *In Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct*, pp. 228–229, 2017.

[3] M. Dikmen and C. M. Burns, “Autonomous driving in the real world: Experiences with tesla autopilot and summon,” *In Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*, pp. 225–228, 2016.

[4] R. Hussain and S. Zeadally, “Autonomous cars: Research results, issues, and future challenges,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2018.

[5] A. G. Ivakhnenko, “Polynomial theory of complex systems,” vol. 1, no. 4, pp. 364–378, 10 1971.

[6] *Cybernetic Predicting Devices*. CCM Information Corporation, 1973.

[7] M. G. Bechtel, E. Mcellhiney, M. Kim, and H. Yun, “Deeppicar: A low-cost deep neural network-based autonomous car,” *2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 11–21, 2018.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[9] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H. Bloecher, K. Werber, and A. Sailer, “Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding,” in *2016 IEEE Radar Conference (RadarConf)*, May 2016, pp. 1–6.

[10] F. P. Muller, E. M. Diaz, and I. Rashdan, “Cooperative positioning and radar sensor fusion for relative localization of vehicles,” in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2016, pp. 1060–1065.

[11] A. A. Ali and H. A. Hussein, “Distance estimation and vehicle position detection based on monocular camera,” in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, May 2016, pp. 1–4.

[12] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang, “Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.*, June 2018, pp. 108–115.

[13] H. Xu, Y. Gao, F. Yu, and T. Darrell, “End-to-end learning of driving models from large-scale video datasets,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, July 2017, pp. 2174–2182.

[14] J. Yang, B. Pelletier, and B. Champagne, “Enhanced autonomous resource selection for lte-based v2v communication,” pp. 1–6, Dec. 2016.

[15] Y. Liu and J. Guo, “Design of improved vehicle collision warning system based on v2v communication,” in *2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, June 2018, pp. 95–98.

[16] K. Sakaguchi and R. Fukatsu, “Cooperative perception realized by millimeter-wave v2v for safe automated driving,” in *2018 Asia-Pacific Microwave Conference (APMC)*, Nov. 2018, pp. 180–182.

[17] S. Wei, Y. Zou, X. Zhang, T. Zhang, and X. Li, “An integrated longitudinal and lateral vehicle following control system with radar and vehicle-to-vehicle communication,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1116–1127, Feb. 2019.

[18] T.-K. Lee, Y.-C. Kuo, S.-H. Huang, G.-S. Wang, C.-Y. Lin, and Y.-C. Tseng, “Augmenting car surrounding information by inter-vehicle data fusion,” pp. 1–6, April 2019.

[19] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, June 2015.

[20] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Proceedings of the 1st Annual Conference on Robot Learning*, 13–15 Nov 2017, pp. 1–16.

[21] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2016.