

# Forward Collision Warning and Lane-mark Recognition Systems Based on Deep Learning

Neng-Sheng Pai,\* Jing-Bin Huang, Jian-Xing Wu, Pi-Yun Chen, and Yue-Han Zhou

Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan (ROC)

(Received December 18, 2019; accepted April 22, 2020)

**Keywords:** deep learning (DL), object recognition, augmented reality (AR), YOLOv2, K-means

In this study, a driver assistance system that uses a network model based on deep learning technology was developed. It has forward collision warning and lane-mark recognition features. The application uses a webcam to capture forward images, which are transferred to a computer in which object recognition has been implemented. The system information is displayed on smart glasses through the network as an augmented reality image. You Only Look Once (YOLO) real-time object detection (tiny YOLOv2) was used as the main architecture to reduce the network complexity and enhance computing efficiency. During the training process, K-means was used to select the anchor box from each dataset. This enabled the size of the predicted box to be determined as a reference to enhance efficiency. This system makes it possible for the driver of a vehicle to learn about the movements and positions of vehicles ahead with respect to distance and lane marks. This reduces the chance of collisions as well as the violations of traffic regulations and improves driving safety.

## 1. Introduction

According to the 2018 World Health Organization (WHO) global road safety report, the number of deaths from road traffic accidents continues to rise. Around 1.35 million people die each year from traffic accidents, and the report emphasizes that road accidents are the main killer of children and adolescents.<sup>(1)</sup> One of the main causes of road accidents arises from drivers paying too little attention to the vehicles on the road in front of them. This may happen because the driver has been distracted or visibility is poor. The problems may also be the fault of the drivers themselves. A vehicle, pedestrian, or even an obstruction may suddenly appear, the driver reacts too slowly, or is unable to take corrective action soon enough, and a collision results. The most common dangerous encounter is with another automobile, a pedestrian, or a motorcycle. Whether it be the driver's fault, or that of a pedestrian or another vehicle, these are all key factors contributing to the sudden appearance of an object in front of a moving vehicle and the cause of accidents. Accidents often result from a violation of regulations such as non-compliance with a traffic sign or traffic lights.

---

\*Corresponding author: e-mail: pai@ncut.edu.tw  
<https://doi.org/10.18494/SAM.2020.2784>

Although many accidents result from intentional violations by drivers, many are the result of the failure of the driver to notice a traffic sign or traffic lights. The problem may be environmental (bad weather) or the result of poor design and inadequate traffic control. In any case, paying too little attention to the road ahead, to traffic signs, and particularly to the vehicle directly in front is dangerous.

The development of technology for autonomous vehicle control has been rapid, and according to the Boston Consulting Group (BCG),<sup>(2)</sup> the size of the global autonomous vehicle market will reach 42 billion US dollars by 2025. The sale of automated vehicles will account for 12.4% of the overall market, and the market scale will double by 2035. At present, the definition of autonomous vehicles in the industry generally complies with the J3016 standard of the Society of Automotive Engineers (SAE). This standard has six levels (0 to 5) depending on the degree of vehicle automation. Autonomous vehicles rely mainly on technologies such as the advanced driver assistance system (ADAS) and the Internet of Vehicles (IoV). For ADAS, forward collision warning (FCW) is the key to solving the problem of a lack of attention to the road ahead and the vehicle in front. Also, road sign recognition (RSR) is a key to the resolution of violations of traffic regulations. However, most research is concerned with standing road signs and traffic lights, and signs on the ground are neglected. Often, too little attention is paid to zebra crossings and yellow box junctions. Failure, for any reason, to observe the instructions given by road-level signs can cause serious accidents that harm both the driver at fault and others on the road.

Early data processing algorithms had drawbacks, namely, features needed to be extracted manually and the computers available could not handle the large amounts of data involved. However, the graphics processing units (GPUs) now freely available, as well as modern fast multicore processors, have raised computing efficiency by orders of magnitude. Artificial intelligence (AI) is now booming and deep learning (DL) has been rejuvenated and is now the most popular AI technology and a serious market focus.

Visual technology is another focus of the entire technology circle and has been a leading trend for years. Major manufacturers have launched augmented reality (AR), virtual reality (VR), and mixed reality (MR) products, and these new technologies have been widely applied in various fields. Spectacular and attractive products, often with special hardware, are available to the public. The market is growing rapidly. According to a forecast by Digi-Capital,<sup>(3)</sup> the AR market will reach 70 to 75 billion US dollars in 2023, and the VR market will reach 10 to 15 billion US dollars. AR technology has huge business potential.

## 2. Related Work

This paper has four parts: DL, object detection, FCW, and lane-mark recognition.

### 2.1 DL

DL is a branch of machine learning (ML). Its algorithms use an artificial neural network (ANN) architecture, which inputs data to implement feature learning. The concept of an ANN can be

traced back to 1943, when a paper by neuroscientist Warren S. McCulloch and mathematician Walter Pitts appeared in the *Bulletin of Mathematical Biology*.<sup>(4)</sup> In this paper, the concept of an ANN was proposed, as well as a mathematical model for artificial neurons. This started the era of research into neural networks, and in 1958, Rosenblatt released the Perceptron,<sup>(5)</sup> the first ANN model, and laid a foundation for serious ANN research. In 1974, Werbos proposed back-propagation (BP)<sup>(6)</sup> to solve the problem of mutual exclusion, which was beyond the capability of basic sensors. In 1986, Rumelhart *et al.* provided a more comprehensive description of BP<sup>(7)</sup> that caused a boom in ANN research. However, a few years later, another BP problem was revealed, the vanishing gradient. This caused stagnation in ANN research until 2006, when Hinton, the father of DL, proposed the restricted Boltzmann machine (RBM)<sup>(8)</sup> and the deep belief network (DBN).<sup>(9)</sup> This solved the vanishing gradient problem, and the deep neural network (DNN) became DL. However, the computing process was run on central processing units (CPUs) at that time. The huge DL calculations strained the capabilities of CPUs, and once again, there was a lull in progress. However, in 2012, at the ImageNet image recognition competition, two of Hinton's students used a GPU plus a deep convolutional neural network (DCNN)<sup>(10)</sup> to win the championship. At that time, GPU operation speed had reached more than 70% of that of CPUs. After this competition, DL became one of the hottest current technologies.

## 2.2 Object detection

There are two types of object detection algorithm: traditional and DL algorithms. The traditional algorithm story began in 2001 when Viola and Jones wrote a thesis regarding object detection,<sup>(11)</sup> which combined three algorithms for facial recognition; they used integral image, adaboost, and the cascade classifier to achieve very good recognition. There are two approaches to DL algorithms. One approach is R-CNN,<sup>(12)</sup> proposed by Girshick *et al.* In this approach, algorithms first generate the candidate regions and then classify them. Although these algorithms have high accuracy, they are slow. The other approach includes You Only Look Once (YOLO)<sup>(13)</sup> proposed by Redmon *et al.*, and SSD<sup>(14)</sup> offered by Liu *et al.* These algorithms predict the location and category probability of the object directly, and although the accuracy is lower, they are much faster than the R-CNN series.

## 2.3 FCW

FCW is an indispensable core technology used in devices for FCW. These include cameras and ultrasonic, radar, and optical sensors. Cameras provide rich image data and can allow the category of a forward object to be identified. This study concentrates on image-based FCW, and many other studies have been made on this aspect in recent years. Among these, Song *et al.*<sup>(15)</sup> used stereo imaging to detect objects in forward vision and also employed UV disparity for image segmentation in the implementation of a FCW system. Mukhtar *et al.*<sup>(16)</sup> selected the sensor first, followed by the detection and tracking of vehicles, and finally provided the best option for a collision avoidance system.

## 2.4 Lane-mark recognition

Some research was done on lane-mark recognition by Gupta and Choudhary,<sup>(17)</sup> and Mathibela *et al.*<sup>(18)</sup> used a single camera to first select a region of interest (ROI) image, and then used grayscale and smoothing before foreground detection and segmentation. The connected components of the processed image were then found using principal component analysis (PCA), and classification was done using spatio-temporal incremental clustering (STIC) to check if it was a traffic lane line or a ground sign. This was followed by graph embedding grassmann discriminant analysis (GGDA) to recognize ground-level signs. Mathibela *et al.*<sup>(18)</sup> focused on the recognition of line-type ground signs. They first divided the signs into seven categories, then used inverse perspective mapping (IPM) to convert the input images into aerial view images. The node position of each image was then determined and used to classify the images. Final classification of the extracted ground road signs was done by random under sampling boost (RUSBoost) and conditional random field (CRF) algorithms.

## 3. Methodology

This section describes Anchor box selection, Tiny YOLO network, and FCW and lane-mark recognition. The system flow chart is shown in Fig. 1.

### 3.1 Anchor box selection

When YOLO was first used, the predicted box was calculated using the results of neural network prediction directly as the box value. However, prediction using this method was inefficient and in YOLOv2, the anchor box was used, mainly to provide a reference of the size of the predicted box, which improved calculation efficiency.

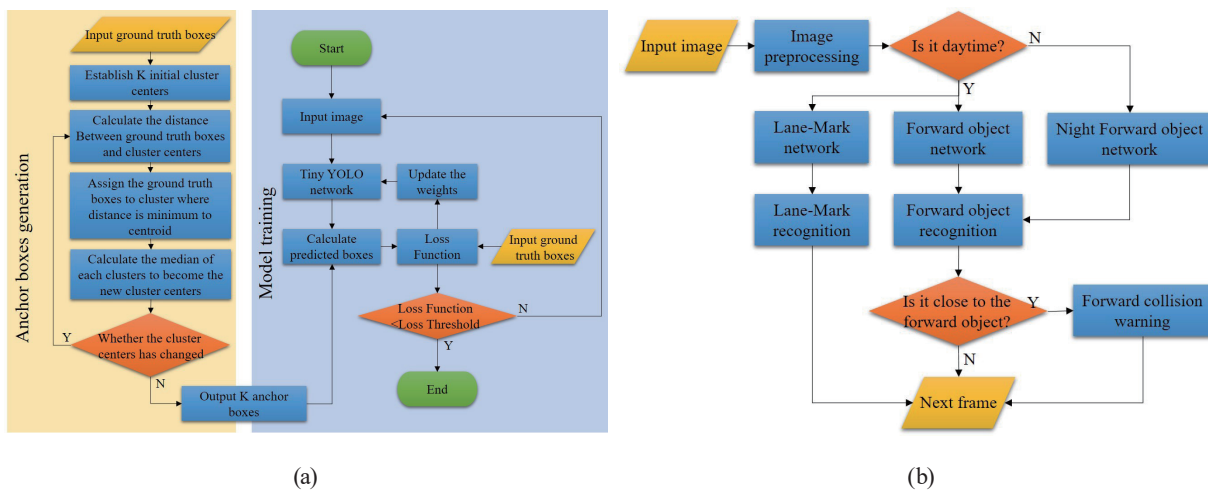


Fig. 1. (Color online) (a) System training process and (b) system detection process.

In this study the K-means algorithm was used for the selection of the anchor box and to classify the required number and size of the anchor boxes from all the ground truth boxes. In addition, since the only anchor box values needed are width and height, the value of the center coordinate is not calculated. The center points of all ground truth boxes are then set and consistent for classification. The steps are shown in the classification flow chart in Fig. 2.

### 3.2 Tiny YOLO network

(1) Network architecture: The network architecture used in this study was based on that of Tiny YOLO. Tiny YOLO was selected to reduce network complexity and enhance computing efficiency, as well as to allow real-time computation. The architecture has nine convolutional layers and six pooling layers, and adds batch normalization (BN)<sup>(19)</sup> in the first eight sets of the convolutional layer and uses Leaky ReLU (LReLU)<sup>(20)</sup> as the activation layer. Figure 3 shows the architecture of the Tiny YOLO network.

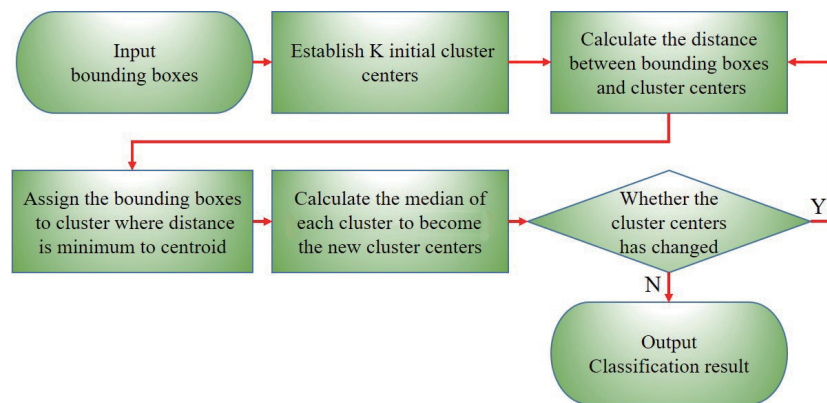


Fig. 2. (Color online) Flow chart—K-means anchor box selection.

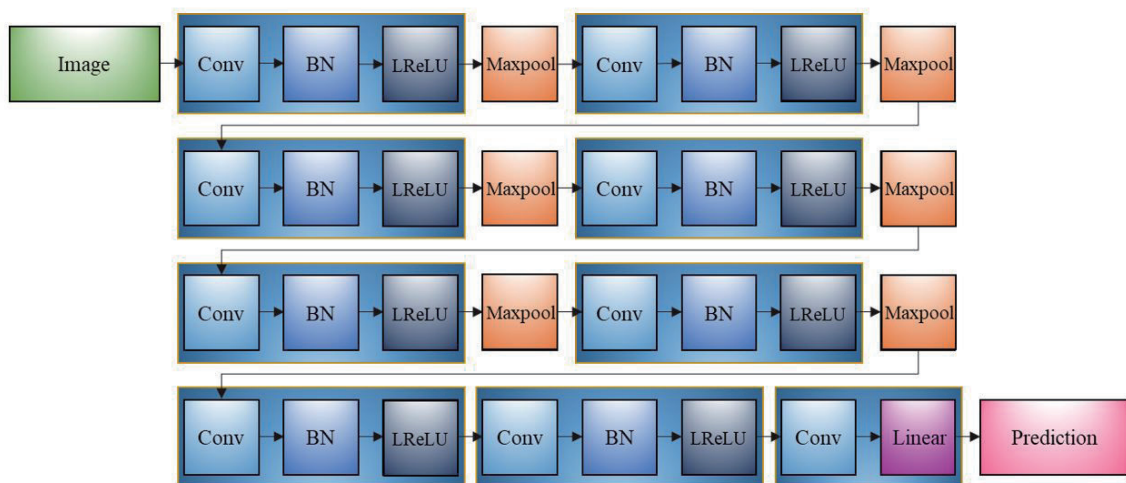


Fig. 3. (Color online) The Tiny YOLO Network.

(2) YOLOv2 principle: The main idea of YOLOv2 is to cut a picture of equal width and height into an  $S \times S$  grid, predict  $B$  pieces of predicted boxes in each grid, and then predict the center coordinates, the size, the confidence, and the probability of  $C$  categories (see Fig. 4).

Equation (1) can be obtained from the above, and the final predicted value can be obtained from Eq. (1).  $S = 13$ ,  $B = 5$ , and  $C = 6$  are used as the settings for the demonstration.

$$P_{final} = S \times S \times B \times (4 + 1 + C) \tag{1}$$

The confidence is defined in Eq. (2), where  $Pr(Object)$  determines whether the predicted box contains the object: if there is no object,  $Pr(Object) = 0$ ; if there is an object,  $Pr(Object) = 1$ . Since the network used in this study was the Tiny YOLO, the accuracy is relatively low compared with other networks; therefore, for the experimental results, the threshold of confidence was set to 0.45.

$$confidence = Pr(Object) \cdot IOU(box_p, box_r)$$

$$Pr(Object) = \begin{cases} 0, & \text{if no object exists in } box_p \\ 1, & \text{if object exists in } box_p \end{cases} \tag{2}$$

In addition,  $IOU(box_p, box_r)$  is the Intersection over Union (IOU) of the predicted box and ground truth box. The calculation method is shown in Eq. (3). In the equation,  $box_p \cap box_r$  is the intersection area of the predicted box and ground truth box, and  $box_p \cup box_r$  is the union area of the predicted and ground truth boxes.

$$IOU(box_p, box_r) = \frac{box_p \cap box_r}{box_p \cup box_r} \tag{3}$$

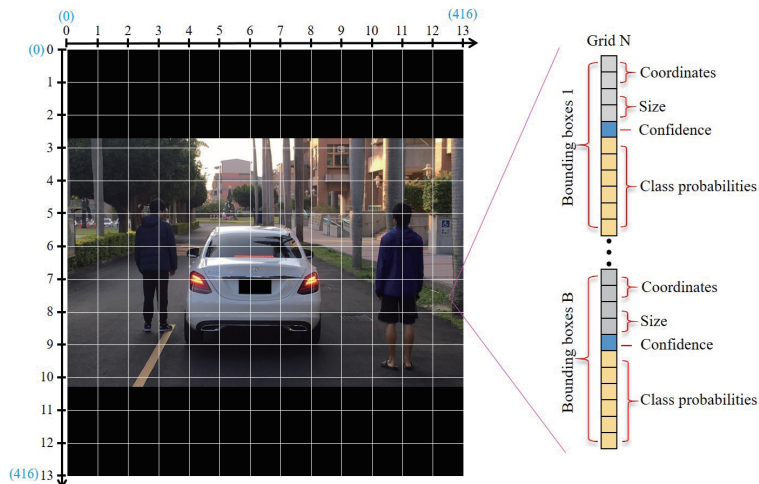


Fig. 4. (Color online) Principle of YOLOv2.



Figure 5 shows a demonstration diagram of the IOU. The green frame is the ground truth box, the blue frame is the predicted box, the area enclosed by the red line is the union, and the area enclosed by the purple line is the intersection.

(3) Loss Function: The main purpose of the Loss Function is to evaluate the difference between the predicted value and the actual value. It is expected that the Loss Function will approach 0 at the end to enable the neural network to have a good predictive effect. The mean square error (MSE) method was used to make the difference between the predicted value and the actual value of the YOLO loss function positive and avoid possible positive and negative value offset.

The Bounding Box Loss Function was divided into four parts: the central coordinate loss, the size loss, the confidence loss, and the category probability loss. Among these, the confidence of the bounding box was also divided into those with an object and those with no object, as shown in Eq. (4).

$$\begin{aligned}
 L = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[ \left( box_{r\_x_{ij}} - box_{p\_x_{ij}} \right)^2 + \left( box_{r\_y_{ij}} - box_{p\_y_{ij}} \right)^2 \right] \\
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[ \left( \sqrt{box_{r_{w_{ij}}}} - \sqrt{box_{p_{w_{ij}}}} \right)^2 + \left( \sqrt{box_{r_{h_{ij}}}} - \sqrt{box_{p_{h_{ij}}}} \right)^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \left( box_{r_{o_{ij}}} - box_{p_{o_{ij}}} \right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{noobj} \left( box_{r_{o_{ij}}} - box_{p_{o_{ij}}} \right)^2 \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \sum_{c \in classes} \left( box_{r\_c_{ijc}} - box_{p\_c_{ijc}} \right)^2
 \end{aligned} \tag{4}$$

### 3.3 FCW and lane-mark recognition

(1) Object recognition: The neural network obtains the predicted box by calculation, but its central coordinates, size, and confidence are not extracted directly from the results. If the

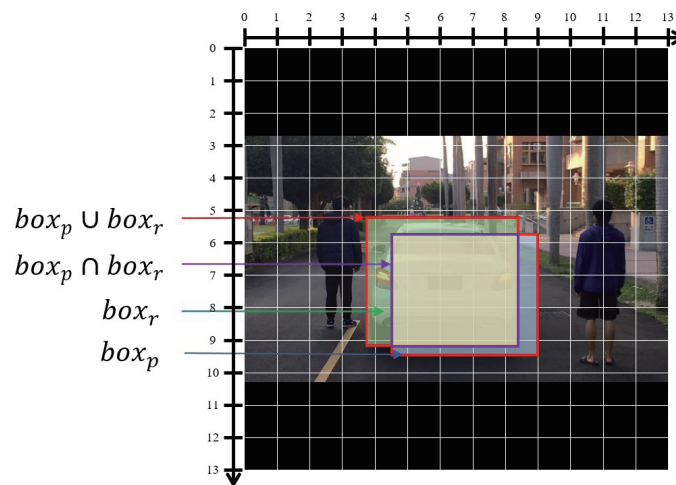


Fig. 5. (Color online) Demonstration diagram of IOU.

results of the prediction are used directly as settings for the predicted box, effective predictions cannot be properly calculated. There are four parts (settings) to the predicted box: the central coordinates, the size, the confidence, and the category probability. A predicted box diagram is shown in Fig. 6.

The first part is the central coordinates of the predicted box.

$$box_{p\_x} = sigmoid(t_x) + c_x \quad (5)$$

$$box_{p\_y} = sigmoid(t_y) + c_y \quad (6)$$

In Eqs. (5) and (6),  $box_{p\_x}$  and  $box_{p\_y}$  are the  $X$  and  $Y$  coordinates of the center of the predicted box, respectively.  $t_x$  and  $t_y$  are the predicted values of the  $X$  and  $Y$  coordinates of the center of the predicted box, respectively.  $c_x$  and  $c_y$  are the  $X$  and  $Y$  coordinates of the upper left of the related block of the current predicted box, respectively. A Sigmoid function is used to limit  $t_x$  and  $t_y$  between 0 and 1 to prevent the coordinates in the box moving to other boxes.

The second part is the size of the predicted box.

$$box_{p\_w} = box_{a\_w} \cdot e^{t_w} \quad (7)$$

$$box_{p\_h} = box_{a\_h} \cdot e^{t_h} \quad (8)$$

In Eqs. (7) and (8),  $box_{p\_w}$  and  $box_{p\_h}$  are the width and height of the predicted box,  $box_{a\_w}$  and  $box_{a\_h}$  are the width and height of the anchor box, and  $t_w$  and  $t_h$  are the predicted width and height of the predicted box, respectively. An exponential function is used to implement scaling to avoid the value being too high. The predicted value is limited to between  $-0.2$  and  $0.2$  via a Gaussian distribution.

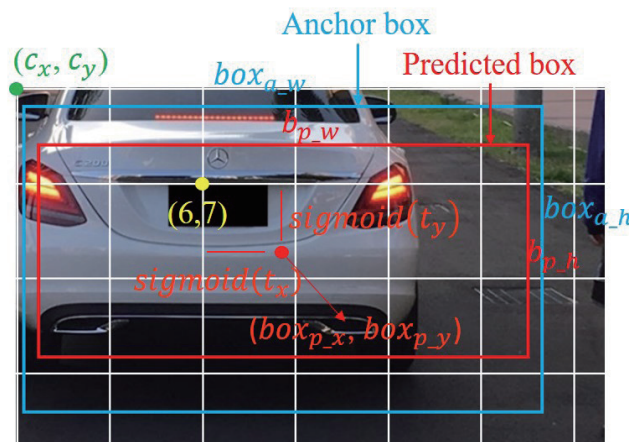


Fig. 6. (Color online) Predicted Box diagram.



The third part is the confidence of the predicted box.

$$box_{p\_o} = sigmoid(t_o) \quad (9)$$

In Eq. (9),  $box_{p\_o}$  is the confidence of the predicted box and  $t_o$  is its confidence predicted value.

The last part is the category probability of the predicted box.

$$box_{p\_c} = t_c \quad (10)$$

In Eq. (10),  $box_{p\_c}$  is the category probability of the predicted box and  $t_c$  is its predicted value.

(2) FCW decision: In addition to object recognition, this system includes FCW to alert the driver to the approach of an object on the road ahead. YOLOv2 can find an object and its position in a picture. The distance between the object's bounding box and the bottom edge of the picture is an indication of how close the object is to the camera. The smaller this distance, the closer the object. When the bounding box is close to the top of the picture, the object is far away. A decision model can be established using this feature as follows.

$$warning = \begin{cases} \text{true,} & \text{if } (object_{btm\_y} > image_h \times 0.75) \\ \text{false,} & \text{otherwise} \end{cases} \quad (11)$$

In Eq. (11),  $object_{btm\_y}$  is the Y-axis coordinate at the bottom of the predicted box for the forward object and  $image_h$  is the height of the picture. When the forward object exceeds the set value, the system will warn the driver. Since this study was carried out in an urban area, the average speed was about 40 to 50 km/h, and a warning was expected when the distance between the object and the camera was 3 to 5 m. The set value used was 75% of the total image height. A schematic diagram of the object collision warning system is shown in Fig. 7. The yellow part is the warning area, and when the image of the object enters this area, the system will send a warning.

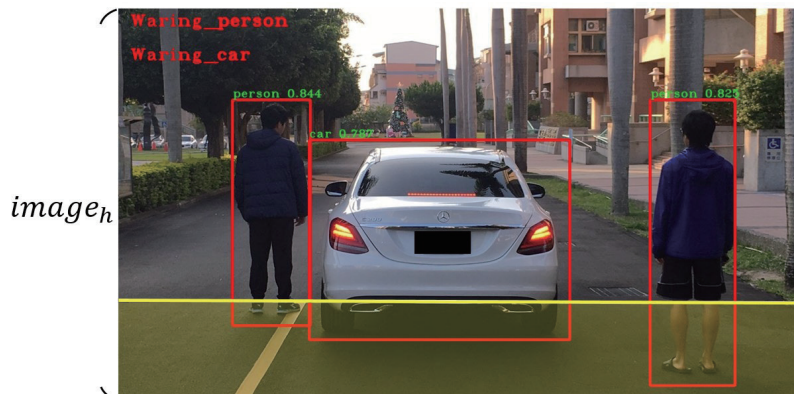


Fig. 7. (Color online) Forward collision decision.

## 4. Experimental

### 4.1 Voice signal preprocessing

Figure 8 shows a hardware architecture diagram, which can be divided into three parts: an input device, a computing device, and a display device. The input device used in this study was a Logitech C525 HD webcam, the computer was a D830MT computing platform with a GTX 1080 Ti GPU for DL, and the AR display device used was BT-350 smart glasses.

### 4.2 Dataset

The dataset used for training in this study was a record of images from the driving recorder. Each image was  $1920 \times 1080$  pixels. Three sets of data were used: a forward object, a night forward object, and a ground road sign. The detailed information is shown in Table 1.

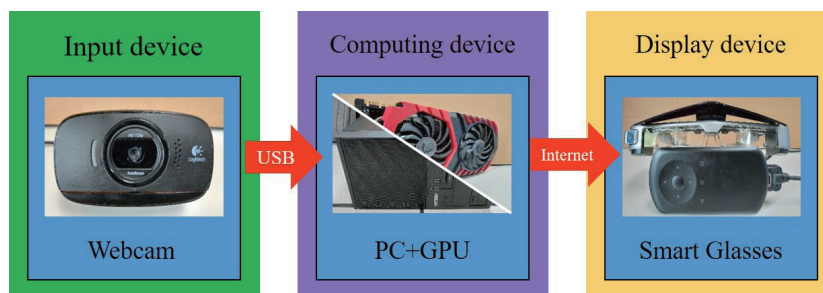


Fig. 8. (Color online) Hardware architecture.

Table 1  
(Color online) Classes and quantity of the dataset.

Dataset name	Number of pictures	Class	Number of bounding boxes
Forward object	546	Person	200
		Bicycle	200
		Motorcycle	300
		Car	300
		Bus	200
		Truck	200
Night forward object	159	Person	100
		Motorcycle	200
		Car	200
Lane-mark	1045	Go straight	200
		Turn left	200
		Turn right	200
		Go straight left	200
		Go straight right	200
		Motorcycle waiting zone	200
		Slow	200
		Crosswalk	200
		Yellow mesh	200

### 4.3 Anchor box setting

K-means was used to calculate three sets of concentrated anchor box data. The results were scaled to  $13 \times 13$  pixels, so that the size of the anchor box matched that of the last layer of the network. The experimental results are shown in Table 2, and the schematics of the prior anchor are shown in Figs. 9–11.

Table 2  
(Color online) Anchor box setting.

Dataset name	Anchor box	Width (pixel)	Height (pixel)
Forward object	Anchor box 1	1.02	0.83
	Anchor box 2	1.6	1.12
	Anchor box 3	2.7	2.08
	Anchor box 4	2.02	4.2
	Anchor box 5	0.71	0.53
	Anchor box 6	0.68	1.6
	Anchor box 7	1.05	2.18
	Anchor box 8	0.46	1.13
	Anchor box 9	4.15	3.61
	Anchor box 10	0.29	0.66
Night forward object	Anchor box 1	0.62	1.69
	Anchor box 2	1.28	0.86
	Anchor box 3	0.33	0.96
	Anchor box 4	2.88	4.97
	Anchor box 5	2.23	1.66
	Anchor box 6	0.74	0.51
Lane-mark	Anchor box 1	3.07	0.18
	Anchor box 2	5.87	0.58
	Anchor box 3	1.69	0.83
	Anchor box 4	10.95	2.66
	Anchor box 5	0.63	0.22
	Anchor box 6	2.67	1.65
	Anchor box 7	1.33	0.49
	Anchor box 8	0.87	0.36
	Anchor box 9	0.374	0.1
	Anchor box 10	1.74	1.44

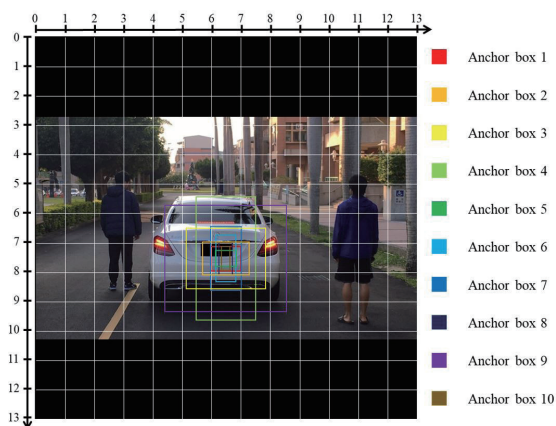


Fig. 9. (Color online) Prior anchor of forward object.

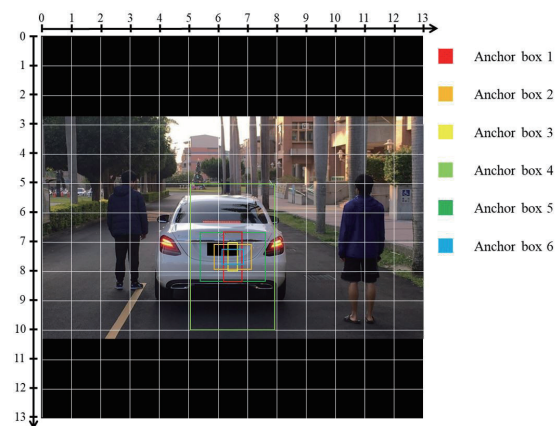


Fig. 10. (Color online) Prior anchor of night forward object.

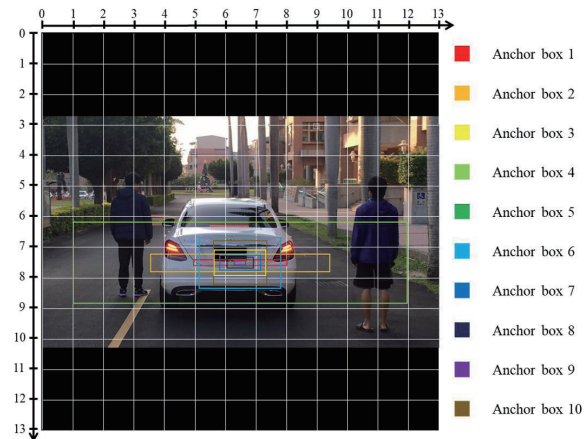


Fig. 11. (Color online) Prior anchor of ground road sign.

Table 3  
(Color online) Parameters of network model.

Type	Size / stride	Filter number	Input size	Output size
Conv1	3×3 / 1	16	416 × 416 × 3	416 × 416 × 16
Maxp1	2×2 / 2		416 × 416 × 16	208 × 208 × 16
Conv2	3×3 / 1	32	208 × 208 × 16	208 × 208 × 32
Maxp2	2×2 / 2		208 × 208 × 32	104 × 104 × 32
Conv3	3×3 / 1	64	104 × 104 × 32	104 × 104 × 64
Maxp3	2×2 / 2		104 × 104 × 64	52 × 52 × 64
Conv4	3×3 / 1	128	52 × 52 × 64	52 × 52 × 128
Maxp4	2×2 / 2		52 × 52 × 128	26 × 26 × 128
Conv5	3×3 / 1	256	26 × 26 × 128	26 × 26 × 256
Maxp5	2×2 / 2		26 × 26 × 256	13 × 13 × 256
Conv6	3×3 / 1	512	13 × 13 × 256	13 × 13 × 512
Maxp6	2×2 / 1		13 × 13 × 512	13 × 13 × 512
Conv7	3×3 / 1	1024	13 × 13 × 512	13 × 13 × 1024
Conv8	3×3 / 1	1024	13 × 13 × 1024	13 × 13 × 1024
Conv9 <sup>(1)</sup>	1×1 / 1	110	13 × 13 × 1024	13 × 13 × 110
Conv9 <sup>(2)</sup>	1×1 / 1	48	13 × 13 × 1024	13 × 13 × 48
Conv9 <sup>(3)</sup>	1×1 / 1	140	13 × 13 × 1024	13 × 13 × 140

Note: Conv = Convolutional, Maxp = Maxpool

#### 4.4 Network model

Three network models based on the Tiny YOLO network were built: a forward object network, a night forward object network, and a ground road sign network. For the Tiny YOLO network, preprocessing is necessary to resize the input image to 416 × 416 pixels. The network model parameters are shown in Table 3, where Conv9<sup>(1)</sup> is the output layer of the forward object network, Conv9<sup>(2)</sup> is the output layer of the night forward object network, and Conv9<sup>(3)</sup> is the output layer of the ground road network.

### 4.5 Training process

Most of the network training was based on the loss function as the end of training. When the loss function reached a certain value, or when the loss curve flattened, the network training stopped. The loss curves of the three groups of network used in this study are shown in Fig. 12.

### 4.6 Recognition results and AR

The system speed used in the FCW and lane-mark recognition experiments was 20 fps. The results of the daytime FCW, nighttime FCW, and lane-mark recognition experiments are respectively shown in Figs. 13–15. In this study, the training dataset and trained network model were used to implement mAP-related experiments, and the experimental results are shown in Table 4.

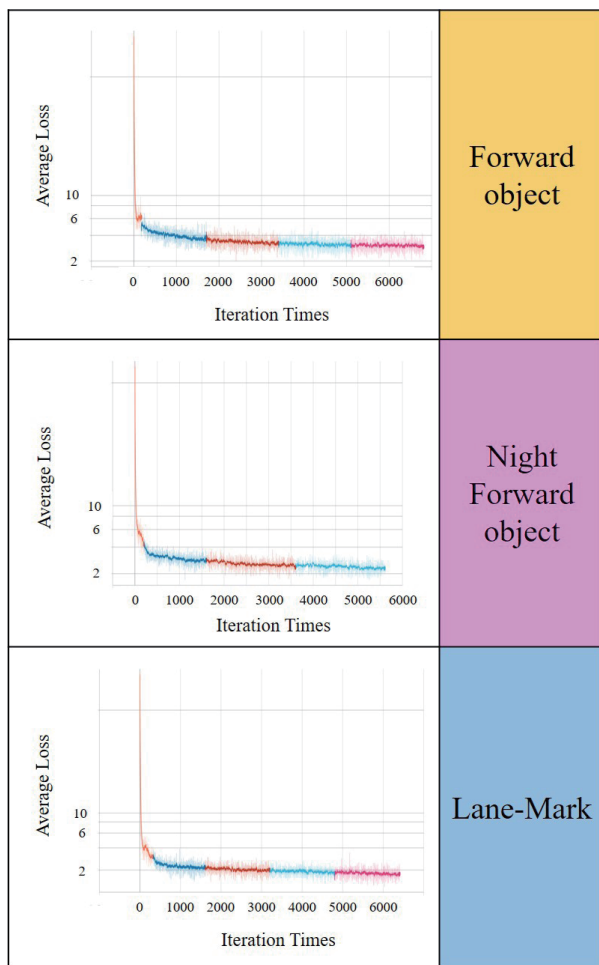


Fig. 12. (Color online) Network Loss curves.



Fig. 13. (Color online) Daytime FCW.

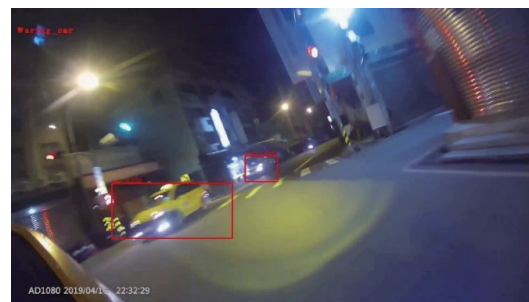


Fig. 14. (Color online) Nighttime FCW.



Fig. 15. (Color online) Lane-mark Recognition.

Table 4  
(Color online) Experimental results.

	Class	Precision	Recall	AP	mAP
Forward object	Motorcycle	0.9321	0.87	0.8445	0.8167
	Car	0.974	0.8733	0.8644	
	Bus	0.9775	0.87	0.8581	
	Bicycle	0.9379	0.83	0.8121	
	Truck	0.9828	0.855	0.8505	
	Person	0.8882	0.715	0.6709	
Night forward object	Car	0.9794	0.95	0.9416	0.806
	Motorcycle	0.8757	0.81	0.7443	
	Person	0.8636	0.76	0.732	
Lane-mark	Turn left	0.7892	0.805	0.7423	0.8028
	Go straight	0.7376	0.745	0.6237	
	Crosswalk	0.78	0.78	0.7131	
	Motorcycle waiting zone	0.92	0.92	0.8979	
	Go straight right	0.9792	0.945	0.9430	
	Turn right	0.8032	0.755	0.686	
	Slow	0.9459	0.875	0.8491	
	Yellow mesh	0.9838	0.91	0.9099	
	Go straight left	0.9514	0.88	0.8603	

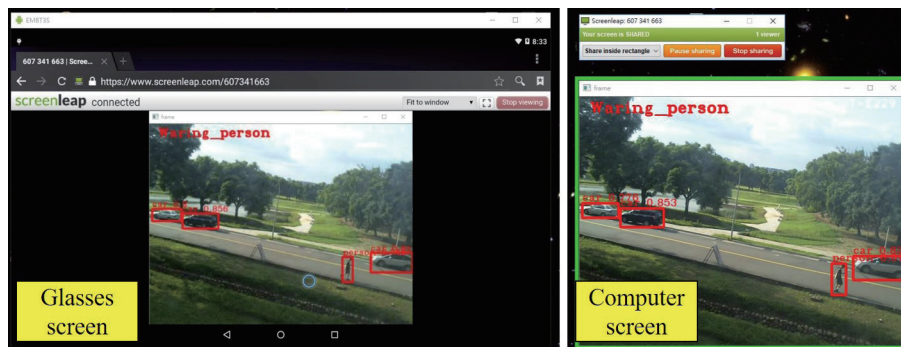


Fig. 16. (Color online) AR test results.

The results from the computer calculations were transmitted to the network via Screenleap and connected to the page with the smart glasses to obtain the current driving image and data from the two systems (Fig. 16).

## 5. Conclusion

The system proposed in this study has two parts with different functions: one is a FCW system and the other is for lane-mark recognition. In these experiments, the data used for training for both forward object and ground RSR were 10 anchor box datasets separately determined by K-means. The two sets of anchor boxes and datasets of the two systems were



used to train the forward object network and lane-mark network models. A third forward object network model was trained for application at night. The anchor box was set to 6, and this improved the performance of nighttime recognition. A camera was used to capture the image of the vehicle ahead and was used to predict the type and position of the forward object and the ground road signs using the three trained network models. The FCW system issues a driver alert if an object is determined to be too close to the camera. The results of the FCW and lane-mark recognition systems were transmitted to BT-350 smart glasses through the Internet by an implementation of AR technology; this gave the driver a view of the road ahead with information about possible hazards and warnings of danger to enhance safety.

### Acknowledgments

This work was supported by the Ministry of Science and Technology, Taiwan, under contract number MOST 108-2221-E-167-025, duration: August 1, 2019–July 31, 2020.

### References

- 1 World Health Organization: Global Status Report on Road Safety 2018, [https://www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2018/en/](https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/) (accessed 28 May 2019).
- 2 Forbes: The Next Car You Buy Could Be Your Last—The Robo-Taxi Era Is Closer Than You Think, <https://www.forbes.com/sites/joannmuller/2017/04/19/the-next-car-you-buy-could-be-your-last-the-robo-taxi-era-is-closer-than-you-think/#6a726a2e21b0> (accessed 13 August 2019).
- 3 Digi-Capital: After Mixed Year, Mobile AR to Drive \$108 Billion VR/AR Market by 2021, <https://www.digi-capital.com/news/2017/01/after-mixed-year-mobile-ar-to-drive-108-billion-vrar-market-by-2021/> (accessed July 20, 2019).
- 4 W. S. McCulloch and W. Pitts: *Bull. Math. Biol.* **5** (1943) 115. <https://doi.org/10.1007/BF02478259>
- 5 F. Rosenblatt: *Psychol. Rev.* **65** (1958) 386. <https://doi.org/10.1037/h0042519>
- 6 P. J. Werbos: PhD Thesis (1974). <https://www.researchgate.net/publication/35657389>
- 7 D. E. Rumelhart, G. E. Hinton, and R. J. Williams: *Nature* **323** (1986) 533. <https://doi.org/10.1038/323533a0>
- 8 G. E. Hinton and R. R. Salakhutdinov: *Science* **313** (2006) 504. <https://doi.org/10.1126/science.1127647>
- 9 G. E. Hinton, S. Osindero, and Y. W. The: *Neural Comput.* **18** (2006) 1527. <https://doi.org/10.1162/neco.2006.18.7.1527>
- 10 A. Krizhevsky, I. Sutskever, and G. E. Hinton: *Adv. Neural Inf. Process. Syst.* **60** (2012) 84. <https://doi.org/10.1145/3065386>
- 11 P. Viola and M. Jones: *Proc. 8th IEEE Int. Conf. Computer Vision 2 (ICCV, 2001)* 747. <https://doi.org/10.1109/ICCV.2001.937709>
- 12 R. Girshick, J. Donahue, T. Darrell, and J. Malik: *IEEE Conf. Computer Vision and Pattern Recognition* (2014) 580. <https://doi.org/10.1109/CVPR.2014.81>
- 13 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: *IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2016)* 779. <https://doi.org/10.1109/CVPR.2016.91>
- 14 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg: *Comput. Vision Pattern Recognit.* (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- 15 W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang: *IEEE Sens. J.* **18** (2018) 5151. <https://doi.org/10.1109/JSEN.2018.2832291>
- 16 A. Mukhtar, L. Xia, and T. B. Tang: *IEEE Trans. Intell. Transp. Syst.* **16** (2015) 2318. <https://doi.org/10.1109/TITS.2015.2409109>
- 17 A. Gupta and A. Choudhary: *IEEE Trans. Intell. Veh.* **3** (2018) 476. <https://doi.org/10.1109/TIV.2018.2873902>
- 18 B. Mathibela, P. Newman, and I. Posner: *IEEE Trans. Intell. Transp. Syst.* **16** (2015) 2072. <https://doi.org/10.1109/TITS.2015.2393715>
- 19 S. Ioffe and C. Szegedy: *Machine Learning* (2015). <https://arxiv.org/abs/1502.03167>
- 20 B. Xu, N. Wang, T. Chen, and M. Li: *Machine Learning* (2015). <https://arxiv.org/abs/1505.00853>