J. Shanghai Jiao Tong Univ. (Sci.), 2021, 26(5): 561-568
https://doi.org/10.1007/s12204-021-2345-x

Camera-Radar Fusion Sensing System Based on Multi-Layer Perceptron

YAO Tong[a] (姚　彤), *WANG Chunxiang*[a] (王春香), *QIAN Yeqiang*[b*] (钱烨强)
(a. Department of Automation; b. University of Michigan - Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: Environmental perception is a key technology for autonomous driving. Owing to the limitations of a single sensor, multiple sensors are often used in practical applications. However, multi-sensor fusion faces some problems, such as the choice of sensors and fusion methods. To solve these issues, we proposed a machine learning-based fusion sensing system that uses a camera and radar, and that can be used in intelligent vehicles. First, the object detection algorithm is used to detect the image obtained by the camera; in sequence, the radar data is preprocessed, coordinate transformation is performed, and a multi-layer perceptron model for correlating the camera detection results with the radar data is proposed. The proposed fusion sensing system was verified by comparative experiments in a real-world environment. The experimental results show that the system can effectively integrate camera and radar data results, and obtain accurate and comprehensive object information in front of intelligent vehicles.
Key words: intelligent vehicle, environmental perception system, sensor fusion, multi-layer perceptron
CLC number: U 471　　**Document code:** A

0 Introduction

Autonomous driving technology can reduce the rate of traffic accidents and provide convenience to people. Therefore, this technology has gradually become the research focus of scholars worldwide. Four important aspects in autonomous driving technology are environmental perception, localization, path planning, and motion control[1]. Among them, the perception and understanding of the environment, that is, the environmental perception technology, is the focus of the autonomous driving system, and the accurate and reliable perception of the environment is the basis for ensuring the safety of intelligent vehicles.

Each sensor has advantages and disadvantages. Thus, the use of a single sensor in an environmental perception system results in the obtainment of limited information. In order to ensure that detailed information about the road ahead can be acquired in real time and accurately, the perception system of intelligent vehicles often adopts multi-sensor fusion to realize the complementation of different sensors[2], thereby obtaining more complete information and improving the ability of the environmental perception system.

Currently, there are two problems with fusion sensing systems. The first is the choice of sensors, which is a prerequisite for the good performance of the fusion sensing system. A reasonable sensor selection is essential to realize the complementarity of advantages and disadvantages of each sensor. The second is the choice of the fusion method. A reasonable fusion method can make full use of the strengths of each sensor, thereby improving the accuracy of the fusion sensing system. In addition, the cost of the sensors must be considered in practical applications.

At present, the commonly used sensors for intelligent vehicles include ultrasonic radar, cameras, radar, light detection and ranging (LiDAR), and infrared detectors. Ultrasonic radar is low in price and simple in data processing, but it is easily affected by temperature and weather with small measurement distance; LiDAR has very high ranging accuracy, strong directionality, and very fast response speed, but the cost is very high, and it is easily affected by the weather; the cost of infrared detectors is very low, but it is also easily affected by the weather, unable to detect distant objects and pedestrians. The image resolution of the camera is high, with rich information. With the rapid development of deep learning in recent years, the accuracy of object detection of the camera has been

Received: 2021-02-05　**Accepted:** 2021-03-05
Foundation item: the National Natural Science Foundation of China (No. U1764264/61873165), and the Shanghai Automotive Industry Science and Technology Development Foundation (No. 1733/1807)
∗**E-mail:** qianyeqiang@sjtu.edu.cn

considerably improved, but various methods based on cameras face difficulties to obtain accurate speed of the object. In contrast, the radar can obtain accurate distance and speed of the object, and has strong robustness to the environment. Therefore, the camera and the radar have complementary advantages and are inexpensive. By combining the results of these two sensors, the category, position, and speed of various traffic objects around the intelligent vehicle can be obtained. Therefore, in this study, the camera and radar were selected to construct the environment perception system of intelligent vehicles.

The output information needs to fuse the information of the two types of sensors chosen: the camera and the radar. In multi-sensor fusion, multiple sensors are first used separately to observe, and then the data obtained by the different sensors are fused according to established rules to obtain more accurate object information. There are three types of multi-sensor fusion: pre-fusion, middle-fusion, and post-fusion.

Pre-fusion refers to the direct fusion of the original data at the input layer; middle fusion refers to the extraction of features from the data returned by each sensor, and fusion of the extracted features; in post-fusion, the original data are preprocessed first, the object features are extracted and identified, and only the identified object is fused. As the data obtained by the camera and radar are not homogeneous, pre-fusion and middle fusion cannot be applied. However, they can be utilized to obtain clear object-level information after preprocessing. Therefore, the post-fusion method was selected in this study.

Regarding the association method of the data of the camera and radar, traditional methods often project radar object data into the image plane. Alessandretti et al.[3] proposed a method for image and radar fusion for vehicle detection. In this method, the object data detected by the radar are projected into the image plane, the object data of radar are used to generate a region of interest on the image, and then vehicle detection is conducted in this area. However, this method has poor accuracy.

Chavez-Garcia and Aycard[4] proposed a method of multi-sensor fusion based on cameras, radar, and LiDAR. In this method, LiDAR is used to screen objects, generate regions of interest in the image, and detect pedestrians in this area. Then, the detected pedestrian and vehicle are fused with the object detected by the radar to distinguish between stationary and moving objects. Kim et al.[5] also proposed a data fusion method based on cameras, radar, LiDAR, and global positioning system (GPS). The methods described above use many sensors, which cause a series of problems, such as joint calibration and time alignment, and also increase the cost.

Pang et al.[6] proposed a camera and LiDAR data fusion method based on a deep learning network, providing a new idea of multimodal fusion, which inspired our idea of the fusion of camera and radar data. However, the 3D detection data obtained by LiDAR are also obtained through a deep learning network and have many available parameters, whereas the data obtained by radar are not processed through a deep learning network and have fewer available parameters. Therefore, using a deep learning network to fuse the data from the camera and radar is not suitable.

Based on the issues discussed above, a multi-layer perceptron (MLP) model for data fusion between camera and radar was proposed in this study, which is more suitable for the problem of fewer available parameters for radar, and whose cost is low. Moreover, a complete environment perception system was constructed for an intelligent vehicle based on a camera and a radar, and a comparison experiment in a real vehicle environment was conducted. The experimental results show that the system can accurately perceive the intelligent vehicle environment.

1 Object Detection of Images

Aiming at the problem of object detection, traditional object detection algorithms include Adaboost[7], histogram of oriented gradients, support vector machine based algorithms[8], and deformable parts model[9]. However, the accuracy of these algorithms is limited. The accuracy of object detection algorithms has been considerably improved since the proposition of the structure of the convolutional neural network (CNN). The classification-based object detection algorithms include R-CNN[10], fast R-CNN[11], and faster R-CNN[12]. These methods extract the feature to select the candidate area from the image input, and then use the classifier and the position to identify the object in the feature space, which is equivalent to performing a two-step operation. Thus, these algorithms have a low speed. The regression-based object detection algorithms include YOLOv1[13], YOLOv2[14], and YOLOv3[15]. These algorithms directly use the entire picture as the input of the network and return the bounding box and the category of the object on the divided grid. Because these algorithms are fast and provide real-time results, they are suitable for autonomous driving. Therefore, in this study, the YOLOv3 algorithm was used for object detection.

The structure of the YOLOv3 object detection algorithm is illustrated in Fig. 1. The Conv2D_BN_Leaky shown in the lower-left corner of the figure refers to the convolution (Conv) + batch normalization (BN) + activation function (Leaky ReLU). In Resblock_Body, n in Res$n$ represents the number of Res-Units contained in this residual block. Each grid unit of YOLOv3 can predict three bounding boxes, and each bounding box can

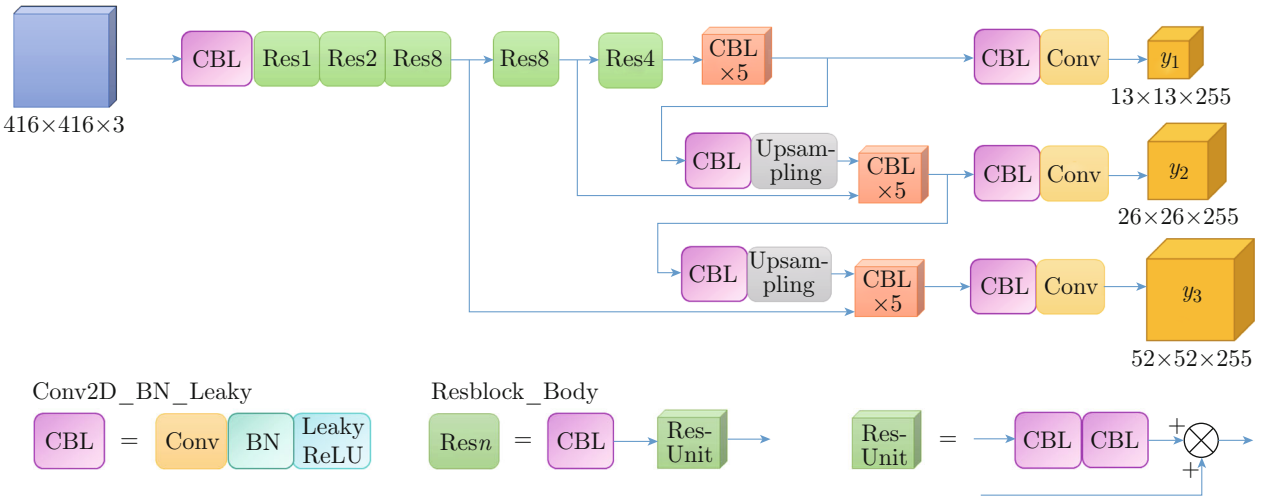

Fig. 1 YOLOv3 structure

predict five parameters: x, y, w, h, and the confidence of the prediction, where (x, y) are the pixel coordinates of the center of the bounding box in the image, w represents the width of the bounding box, and h represents the height of the bounding box. In addition, YOLOv3 can predict the probability of 80 categories; thus, the output size of the tensor is $3 \times (5 + 80) = 255$.

YOLOv3 is trained end-to-end, and its loss function is shown in

$$
\begin{aligned}
L = {} & \lambda_{\text{crd}} \sum_{i=0}^{S_{\text{G}}^2} \sum_{j=0}^{N_{\text{b}}} I_{ij}^{\text{obj}} [(x_i^j - \hat{x}_i^j)^2 + (y_i^j - \hat{y}_i^j)^2] + \\
& \lambda_{\text{crd}} \sum_{i=0}^{S_{\text{G}}^2} \sum_{j=0}^{N_{\text{b}}} I_{ij}^{\text{obj}} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] - \\
& \sum_{i=0}^{S_{\text{G}}^2} \sum_{j=0}^{N_{\text{b}}} I_{ij}^{\text{obj}} [\hat{C}_i^j \log C_i^j + (1 - \hat{C}_i^j) \log (1 - C_i^j)] - \\
& \lambda_{\text{nbj}} \sum_{i=0}^{S_{\text{G}}^2} \sum_{j=0}^{N_{\text{b}}} I_{ij}^{\text{nbj}} [\hat{C}_i^j \log C_i^j + (1 - \hat{C}_i^j) \log (1 - C_i^j)] - \\
& \sum_{i=0}^{S_{\text{G}}^2} I_{ij}^{\text{obj}} \sum_{c \in \mathcal{C}} [\hat{P}_i^j \log P_i^j + (1 - \hat{P}_i^j) \log (1 - P_i^j)]. \qquad (1)
\end{aligned}
$$

In Eq. (1), λ_{crd} represents the weight of localization error; λ_{nbj} represents the weight of classification error; S_{G} represents the grid size; N_{b} represents the number of anchor boxes in each grid; I_{ij}^{obj} represents whether the jth anchor box of the ith grid is responsible for this object, if responsible for $I_{ij}^{\text{obj}} = 1$, otherwise $I_{ij}^{\text{obj}} = 0$; I_{ij}^{nbj} represents whether the jth anchor box of the ith grid is not responsible for this object; $\hat{x}_i^j$ and x_i^j represent the predicted and true x-coordinates, respectively; $\hat{y}_i^j$ and y_i^j represent the predicted and true y-coordinates, respectively; $\hat{w}_i^j$ and w_i^j represent the predicted and true bounding box widths, respectively; $\hat{h}_i^j$ and h_i^j represent the predicted and true bounding box heights, respectively; $\hat{C}_i^j$ and C_i^j represent the predicted confidence and the true confidence, respectively; $\hat{P}_i^j$ and P_i^j represent the predicted and true category probabilities, respectively; c represents the object category obtained by image detection, and $\mathcal{C}$ is the classes. After training, the value of the total loss function is minimized.

In this study, the YOLOv3 object-detection algorithm was used for detection. The used camera is the AXIS vehicle-mounted front-view camera. The detection result of a test conducted on a highway is shown in Fig. 2.

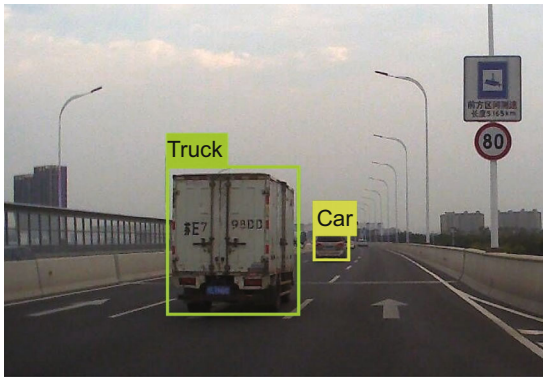

Fig. 2 Object detection

2 Preprocessing of Radar Data

Because the radar and camera are not in the same coordinate system, the radar data cannot be directly combined with the camera data. Therefore, coordinate transformation was applied to transform the data from the radar coordinate system to the camera coordinate system. In practical applications, the position

and posture are usually combined into a coordinate system. The relative position between the two coordinate systems can be described by a translation vector plus a rotation matrix. The rotation matrix is used to describe the posture of the coordinate system relative to the reference system, and the translation vector is used to describe the translation of the origin of the coordinate system relative to the origin of the reference system. When translation and rotation transformations occur simultaneously, suppose that the vector determining the origin of the coordinate system $\{B\}$ is ${}^{A}\boldsymbol{P}_{\mathrm{BORG}}$, and the rotation matrix of $\{B\}$ relative to the reference system $\{A\}$ is ${}^{A}_{B}\boldsymbol{R}$, as shown in Fig. 3.

Here, the transformation relationship from a coordinate system $\{B\}$ to a reference coordinate system $\{A\}$ is used as follows:

$$\begin{bmatrix} {}^{A}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}\boldsymbol{R} & {}^{A}\boldsymbol{P}_{\mathrm{BORG}} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}^{B}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix}. \tag{2}$$

As it is complicated to use the rotation matrix to represent the posture, another representation method is often used in practical applications, which only requires three elements. As shown in Fig. 4, the coordinate system $\{B\}$ is set to coincide with the reference coordinate system $\{A\}$; $\{B\}$ is first rotated around an axis $\hat{x}_A$ by angle γ (roll angle), rotated around an axis $\hat{y}_A$ by angle β (pitch angle), and finally rotated around the $\hat{z}_A$ axis by an angle α (yaw angle), where α, β, and γ are also called Euler angles.

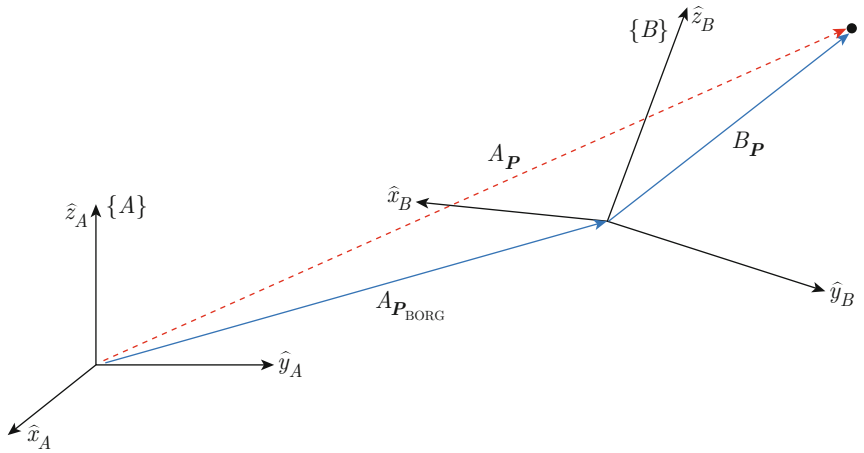

Fig. 3 Coordinate system translation and rotation transformations

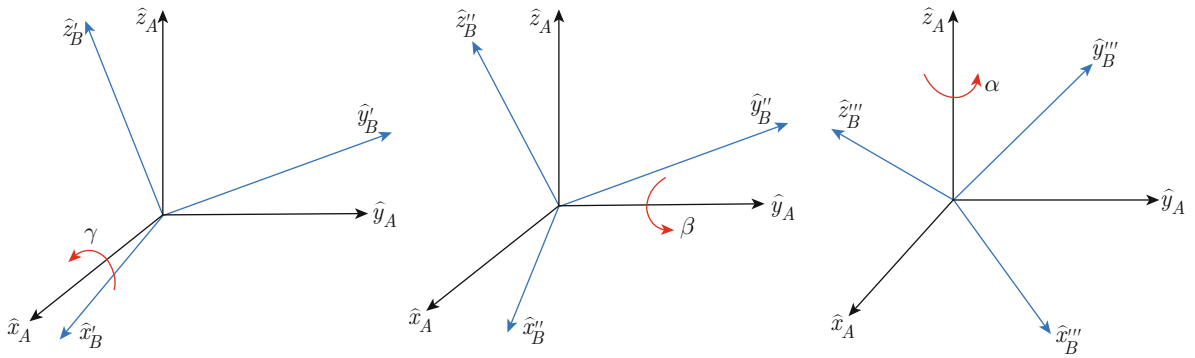

Fig. 4 Euler angles

The coordinate system transformation relationship of this method is expressed as a rotation matrix:

$${}^{A}_{B}\boldsymbol{R}_{xyz}(\alpha,\gamma,\beta) = \begin{bmatrix} \mathrm{c}\alpha\mathrm{c}\beta & \mathrm{c}\alpha\mathrm{s}\beta\mathrm{s}\gamma - \mathrm{s}\alpha\mathrm{c}\gamma & \mathrm{c}\alpha\mathrm{s}\beta\mathrm{c}\gamma + \mathrm{s}\alpha\mathrm{s}\gamma \\ \mathrm{s}\alpha\mathrm{c}\beta & \mathrm{s}\alpha\mathrm{s}\beta\mathrm{s}\gamma + \mathrm{c}\alpha\mathrm{c}\gamma & \mathrm{s}\alpha\mathrm{s}\beta\mathrm{c}\gamma - \mathrm{c}\alpha\mathrm{s}\gamma \\ -\mathrm{s}\beta & \mathrm{c}\beta\mathrm{s}\gamma & \mathrm{c}\beta\mathrm{c}\gamma \end{bmatrix}, \tag{3}$$

where $\mathrm{c}\alpha$ is the abbreviation of $\cos\alpha$, $\mathrm{s}\alpha$ is the abbreviation of $\sin\alpha$, and the other parameters follow this definition.

In practical applications, in which the inverse solution is often applied, that is, using a rotation matrix, the equivalent Euler angle relationship is derived. A known rotation matrix is

$${}^{A}_{B}\boldsymbol{R}_{xyz}(\alpha,\beta,\gamma) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \tag{4}$$

where α, β, and γ can be calculated by

$$\left.\begin{aligned}\alpha &= \arctan 2(r_{21}/\mathrm{c}\beta, r_{11}/\mathrm{c}\beta)\\ \beta &= \arctan 2(-r_{31}, \sqrt{r_{11}^2+r_{21}^2})\\ \gamma &= \arctan 2(r_{32}/\mathrm{c}\beta, r_{33}/\mathrm{c}\beta)\end{aligned}\right\}. \tag{5}$$

In this study, through joint calibration, a transformation matrix from the radar to the camera was obtained. Then, the transformation matrix was expressed in the form of Euler angles and the translation parameters from the x, y and z axes, and the inverse calculation was performed using Eq. (5) to obtain α, β, γ, and the translation parameters.

The data returned by the radar are in the form of a set of arrays, the readability is relatively poor, and information cannot be obtained intuitively. In this study, the radar data were preprocessed and transformed to the camera coordinate system. The process of preprocessing is shown in Fig. 5.

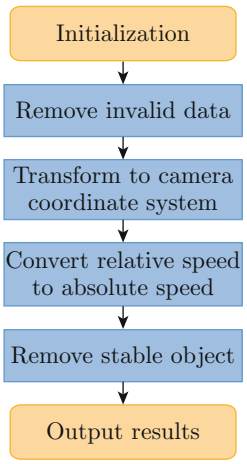

Fig. 5 Preprocessing process

The Delphi forward-looking side radar was used in this study. After receiving the object data of the radar, the position information of the obtained object is expressed as a position vector relative to the radar:

$$\boldsymbol{P}_{\mathrm{o}} = [P_{ox} \quad P_{oy} \quad 0 \quad 1]^{\mathrm{T}}, \tag{6}$$

where (P_{ox}, P_{oy}) are the coordinates of the detected object. After transform to the camera coordinate system, the new obtained position vector is expressed as

$$\boldsymbol{P}_{\mathrm{t}} = [P_{tx} \quad P_{ty} \quad 0 \quad 1]^{\mathrm{T}}, \tag{7}$$

where (P_{tx}, P_{ty}) are the object coordinates after transform. A schematic of P_{tx} and P_{ty} is shown in Fig. 6.

The radar can also determine the speed of an object. Because the radar is installed on the car, the obtained speed is the relative speed v_{r} of the object relative to the car, which should be converted to absolute speed v_{a}, as follows:

$$v_{\mathrm{a}} = v_{\mathrm{r}} - v_{\mathrm{c}}, \tag{8}$$

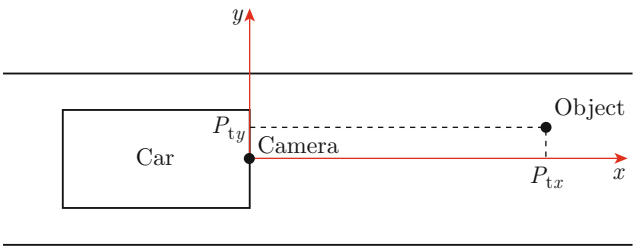

Fig. 6 Schematic diagram of top view

where v_{c} is the speed of the car. Then, the preprocessed data result O_{r} of each object is the output in the form of an array:

$$O_{\mathrm{r}} = \{P_{tx}, P_{ty}, v_{\mathrm{a}}\}. \tag{9}$$

After the described processing, the radar data can be converted from the radar coordinate system to the camera coordinate system.

3 MLP Model for Camera and Radar Data Association

An MLP is a common model in machine learning. It usually adds one or several hidden layers between the input and output layers. Each hidden layer has several units to learn the features better. The general structure of an MLP is shown in Fig. 7. After each layer, an excitation function is typically used to learn nonlinear features. The commonly used activation functions include ReLU, sigmoid function, and tanh.

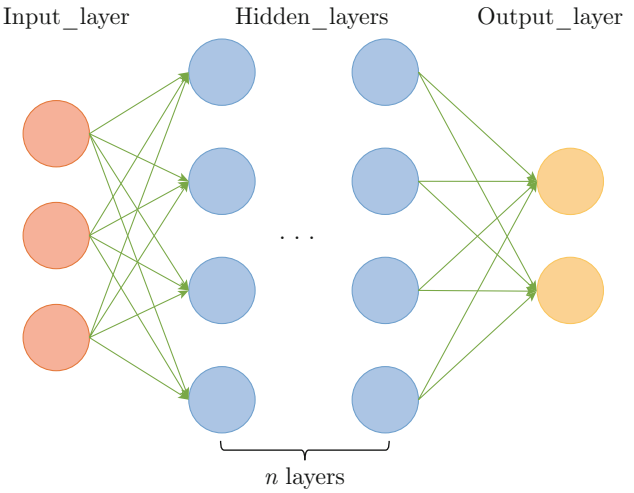

Fig. 7 MLP model

Backpropagation is often used to update the model parameters, and refers to defining a loss function for the output layer, minimizing the loss function, adopting a gradient descent method, and updating the parameters of each layer model, as expressed in

$$W_l := W_l - \eta \frac{\partial E}{\partial W_l}, \tag{10}$$

where W_l represents the parameter value of each layer, E represents the loss function for the output layer, and η represents the learning rate.

In the data association model in this study, the input is $(x_{\text{cen}}, y_{\text{cen}}, x_{\text{rad}}, y_{\text{rad}})$, where $(x_{\text{cen}}, y_{\text{cen}})$ are the pixel center coordinates of the camera object detection, and $(x_{\text{rad}}, y_{\text{rad}})$ are the coordinates of the object detected by the radar. There are two hidden layers in the middle, and each hidden layer contains four units. The activation function is ReLU, as shown in $\phi(X) = \max(0, X)$. The output layer has one unit. In order to determine whether the camera detection object and the radar detection object are the same, the output layer uses the sigmoid function to limit the output value to the range of [0, 1], as shown in $S(X) = [1+\exp(-X)]^{-1}$. The loss function uses the binary cross-entropy. The structure of the MLP model used in this study is shown in Fig. 8.

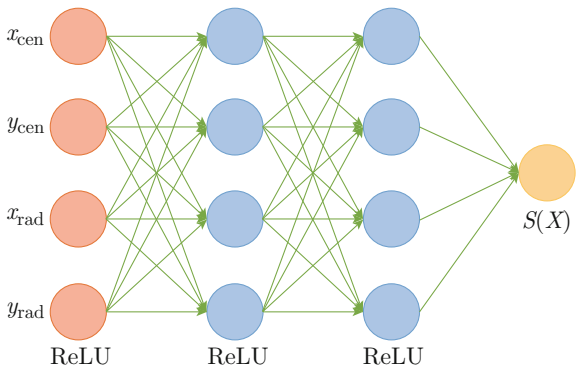

Fig. 8 Network model used in this study

The training dataset of the model was collected in the campus environment of Shanghai Jiao Tong University, and the positive and negative samples were evenly distributed. The accuracy of the final model on the training dataset was 97.5%, and the accuracy of the validation dataset was 93.75%. The model training process is shown in Fig. 9, where a represents the accuracy and t is the number of iterations.

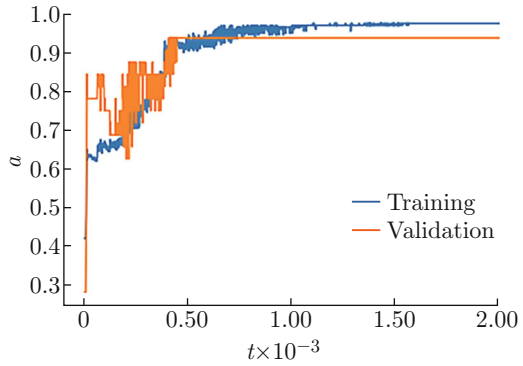

Fig. 9 Model training process

4 Fusion Perception System and Experiment

The structure of the fusion perception system built in this study is shown in Fig. 10. It communicates through the message subscribing and publishing of the robot operate system (ROS). For ROS, the information of the camera object detection and the information obtained after the preprocessing of the radar are published, the two types of information are received, and then the fusion model is called for judgement. If it is considered to be the same object, the camera object detection data are associated with the radar data, and the information of the position, speed and category of the object is output. The format of the output information is

$$O_{\text{PSC}} = \{c, x_{\text{bd}}, y_{\text{bd}}, P_{tx}, P_{ty}, v_{\text{a}}\}, \tag{11}$$

where $(x_{\text{bd}}, y_{\text{bd}})$ represent the center coordinates of the bounding box obtained by the object detection algorithm.

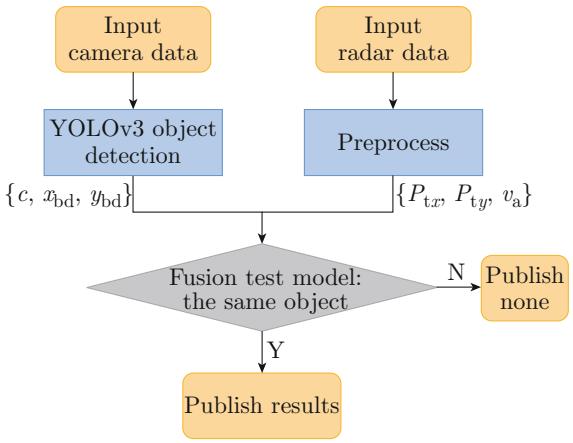

Fig. 10 Structure of the fusion sensing system

The CyberTiggo autonomous driving platform was used and a real-world experiment was conducted at Shanghai Jiao Tong University and highway. The results of the fusion perception system are presented in Fig. 11.

In the actual application of the camera and radar data association, the method of projecting the radar data to the pixel coordinate system and using a static threshold for judgment is often adopted. In this study, comparative experiments were conducted using the two methods based on projection.

The first comparison method (Comparison 1) consisted of projecting the radar data to the pixel coordinate system through the external parameter matrix, internal parameter matrix, and distortion coefficients, and then determining whether the projection point is in the bounding box of the detected object. If it is in the bounding box, it has the same goal. The external parameter matrix is the transformation matrix from the

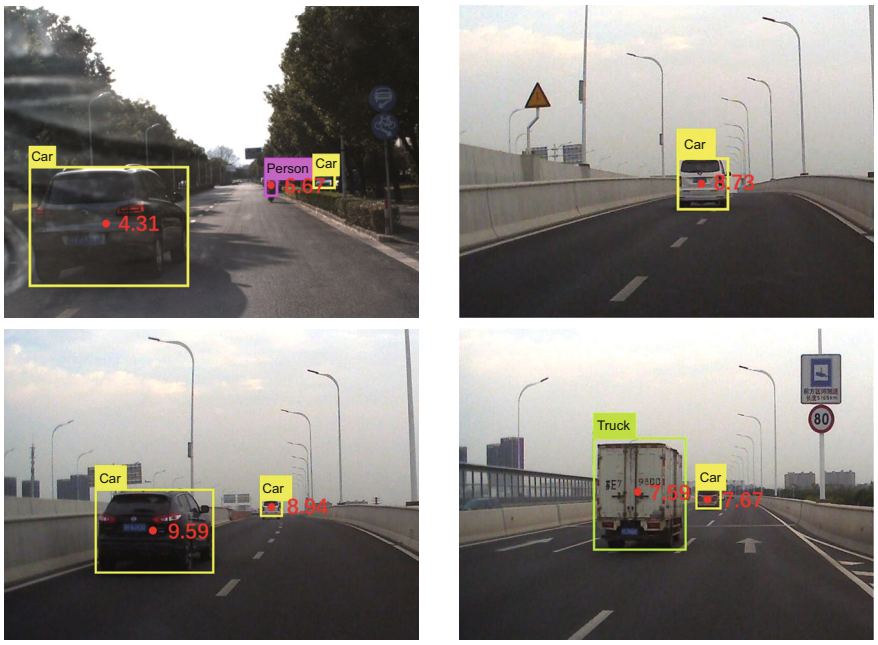

Fig. 11 Results of the fusion perception system (speed values in m/s)

radar to the camera coordinate system, as mentioned previously. The internal parameter matrix represents the relationship between the camera and the pixel coordinate systems. The distortion coefficients include radial and tangential distortions, which can be obtained by camera calibration.

The second comparison method (Comparison 2) consisted of projecting the radar data to the pixel coordinate system, adopting the method of joint Gaussian probability distribution, and setting a threshold. If the probability of the center of the bounding box to be within the probability distribution is higher than this threshold, then the object is the same. The joint Gaussian probability distribution is given by

$$P(\boldsymbol{b}_{\mathrm{c}}) = \frac{\exp\left(-\dfrac{(\boldsymbol{b}_{\mathrm{c}}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{b}_{\mathrm{c}}-\boldsymbol{\mu})}{2}\right)}{(\sqrt{2\pi})^{n}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}, \tag{12}$$

where

$$\begin{aligned}
&\boldsymbol{b}_{\mathrm{c}} = [x_{\mathrm{bd}} \quad y_{\mathrm{bd}}]^{\mathrm{T}},\\
&\boldsymbol{b}_{\mathrm{c}} \sim \mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}),\\
&\boldsymbol{\mu} = [\mu_x \quad \mu_y]^{\mathrm{T}},
\end{aligned}$$

(μ_x, μ_y) are the coordinates of the radar data projected to the pixel coordinate system, and $\boldsymbol{\Sigma}$ is a positive definite matrix of 2×2, representing the covariance.

The accuracy of the three methods was determined, as shown in Table 1.

By analyzing the results of the three methods, it can be observed that the first comparison method has a very good effect on the nearby target with a large bounding box. It ensures that the projection point of the radar data is projected onto the bounding box. However, for distant objects with a small bounding box, the projection points of radar data can be easily projected outside the bounding box owing to errors in joint calibration, which leads to missed detection.

Table 1 Accuracy of the methods evaluated

Method	a/%
Ours	97.5
Comparison 1	96.0
Comparison 2	92.5

In the second comparison method, the effect of using the probability distribution to set the static probability threshold is highly dependent on the choice of threshold. An excessively large threshold would result in several missed detections, whereas a small threshold would result in several false detections. For near and far objects, the suitable thresholds are evidently different. Camera and radar are more accurate in detecting nearby objects and are more suitable for smaller thresholds. The detection error for distant objects is larger, and they are more suitable for larger thresholds.

The advantage of the method proposed in this study for fusion judgment using the MLP model is that it does not require the setting of a threshold. The training process of the MLP can automatically adjust the required parameters based on the results to avoid the problem caused by the selection of the threshold, and the accuracy of the judgment is improved.

5 Conclusion

Multi-sensor fusion is often used in the environmental perception systems of intelligent vehicles. However, multiple sensors face problems such as the choice of sensors and fusion methods. In order to solve these problems, a machine learning based fusion sensing system that uses a camera and a radar was proposed, which can be used in intelligent vehicles. First, the YOLOv3 object detection algorithm was used to detect the image obtained by the camera, convert the obtained radar data to the camera coordinate system, and then perform preprocessing. Second, an MLP model for correlating camera detection results and radar data was proposed to obtain accurate and comprehensive object information in front of intelligent vehicles. Finally, the proposed fusion perception system was verified through a comparative experiment in a real-world environment. The experimental results show that compared with other methods, the method proposed in this study can accurately fuse the camera and radar results. The proposed environmental perception system in this study can be further improved by collecting more data to improve the model in the proposed environmental perception system, conducting more real-world experiments to verify the generalization of the model, and adding more sensors to the environmental perception system to improve its perception ability.

References

[1] YURTSEVER E, LAMBERT J, CARBALLO A, et al. A survey of autonomous driving: Common practices and emerging technologies [J]. *IEEE Access*, 2020, **8**: 58443-58469.

[2] FAYYAD J, JARADAT M A, GRUYER D, et al. Deep learning sensor fusion for autonomous vehicle perception and localization: A review [J]. *Sensors* (*Basel, Switzerland*), 2020, **20**(15): E4220.

[3] ALESSANDRETTI G, BROGGI A, CERRI P. Vehicle and guard rail detection using radar and vision data fusion [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2007, **8**(1): 95-105.

[4] CHAVEZ-GARCIA R O, AYCARD O. Multiple sensor fusion and classification for moving object detection and tracking [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2016, **17**(2): 525-534.

[5] KIM B, KIM D, PARK S, et al. Automated complex urban driving based on enhanced environment representation with GPS/map, radar, lidar and vision [J]. *IFAC-PapersOnLine*, 2016, **49**(11): 190-195.

[6] PANG S, MORRIS D, RADHA H. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection [C]//*2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Las Vegas, NV: IEEE, 2020: 10386-10393.

[7] VIOLA P, JONES M J. Robust real-time face detection [J]. *International Journal of Computer Vision*, 2004, **57**(2): 137-154.

[8] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C]//*2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, CA: IEEE, 2005: 886-893.

[9] FELZENSZWALB P F, GIRSHICK R B, MCALLESTER D, et al. Object detection with discriminatively trained part-based models [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, **32**(9): 1627-1645.

[10] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//*2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH: IEEE, 2014: 580-587.

[11] GIRSHICK R. Fast R-CNN [C]//*2015 IEEE International Conference on Computer Vision*. Santiago: IEEE, 2015: 1440-1448.

[12] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **39**(6): 1137-1149.

[13] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]//*2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV: IEEE, 2016: 779-788.

[14] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger [C]//*2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI: IEEE, 2017: 6517-6525.

[15] REDMON J, FARHADI A. YOLOv3: An incremental improvement [C]//*2018 IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, 2018: 2513-2520.