Chinese Journal of Mechanical Engineering

ORIGINAL ARTICLE

Open Access

Deep Learning Based Data Fusion for Sensor Fault Diagnosis and Tolerance in Autonomous Vehicles

Huihui Pan[1,2], Weichao Sun[1], Qiming Sun[1] and Huijun Gao[1*]

Abstract

Environmental perception is one of the key technologies to realize autonomous vehicles. Autonomous vehicles are often equipped with multiple sensors to form a multi-source environmental perception system. Those sensors are very sensitive to light or background conditions, which will introduce a variety of global and local fault signals that bring great safety risks to autonomous driving system during long-term running. In this paper, a real-time data fusion network with fault diagnosis and fault tolerance mechanism is designed. By introducing prior features to realize the lightweight network, the features of the input data can be extracted in real time. A new sensor reliability evaluation method is proposed by calculating the global and local confidence of sensors. Through the temporal and spatial correlation between sensor data, the sensor redundancy is utilized to diagnose the local and global confidence level of sensor data in real time, eliminate the fault data, and ensure the accuracy and reliability of data fusion. Experiments show that the network achieves state-of-the-art results in speed and accuracy, and can accurately detect the location of the target when some sensors are out of focus or out of order. The fusion framework proposed in this paper is proved to be effective for intelligent vehicles in terms of real-time performance and reliability.

Keywords: Autonomous vehicles, Fault diagnosis and tolerance, Object detection, Data fusion

1 Introduction

Road object detection is one of the core technologies of autonomous vehicles. It provides real-time information on road elements such as surrounding vehicles and pedestrians for autonomous vehicles in real time. Traditional object detection technology is based on environmental perception sensors like cameras. The camera can obtain rich color and contour information from the environment, which facilitates the machine learning algorithm to find the object of interest and judge its category.

In the 1990s, road object detection tasks were mainly completed by artificial features combined with machine learning [1]. Artificial features were widely used at that time, such as HOG, LBP, and Haar features. A set of feature vectors will be obtained after extracting artificial features in local areas of the image. Using feature methods, such as SVM to classify feature vectors, the location and category of target objects can be detected. With the development of deep learning, the object detection tasks based on image sensors are mainly completed by convolutional neural networks. Krizhevsky presents AlexNet [2] which reduces the Top-5 error rate of classification task to 15.3%. ResNet devides the network into several blocks whose input and output are directly connected through a Shortcut structure [3], which effectively solves the problem of gradient disappearance. To solve the problem that traditional object detection algorithms need to traverse sensor data for many times and thus cause serious delay, Girshick et al. propose RCNN network [4–6], which uses three stages to complete object detection: feature extraction backbone, region proposal

*Correspondence: hjgao@hit.edu.cn
[1] Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China
Full list of author information is available at the end of the article

network (RPN) and RoI-Pooling. Inspired by RCNN, researchers begin to optimize the performance of object detection for specific traffic environment. Murugan et al. [7, 8] improve the structure of RCNN by using frame difference information of traffic surveillance video to train the network and monitor the moving vehicles in the video in real-time. Ref. [9] uses Faster-RCNN combined with selective search method to detect forward vehicles, which solves the problem of vehicle loss to a large extent. Qu et al. [10] use CNN to process UAV data, focusing on identification of distant moving vehicles and achieving high accuracy. To solve the problem of poor real-time performance of the multi-stage model, YOLO [11–13] and SSD [14] are proposed to use the one-stage network to complete the bounding box regression and classification at the same time, and increase the object detection speed to 20 frames per second (fps).

To reduce the pressure on the computing and storage unit of autonomous vehicles, researchers are devoted to processing multi-source environmental perception data through data fusion methods, the purpose of which is to comprehensively use the advantages of each sensor carried by autonomous vehicles to obtain object detection results that are better than any single sensor. Generally, the data fusion method can be divided into pre-fusion method and post-fusion method according to the locations where fusion occurs. The former fuses the sensor data in the original input layer, and then design the object detection network to process on the fusion data [15–18]. Compared with the pre-fusion algorithm, the fusion position and strategy of post-fusion algorithm are more flexible and variable. It usually performs feature extraction (FE) on each sensor data, and then designs specific fusion strategies with specific tasks [19–21].

The sensor will inevitably introduce fault signals during long-term running. Therefore, the stability of the algorithm [22], as well as the ability of fault diagnosis and tolerance are very important to the safe driving of autonomous vehicles. The work in Ref. [23] presents a novel penalty domain selection machine (PDSM) enabled transfer learning for gearbox fault recognition (GFR), which is an effective tool to solve the real GFR problem under insufficient data condition. Yan et al. propose a faster and more accurate deep learning framework for highly accurate machine fault diagnosis using transfer learning and achieved state-of-the-art results in main mechanical datasets [24], which proves that the transfer learning can enable and accelerate the training of the deep neural network with high accurate.

In this paper, we utilize a different method to complete the object detection task, and concentrate on solving the problems that are not considered in the previous works. Specifically, we propose a novel data fusion framework with a lightweight structure, which can process large-scale multi-modal data in real time. Importantly, in view of the sensor fault problems that may occur during driving, this paper proposes a fault diagnosis and avoidance (FDA) mechanism in the data fusion framework, and conducts mutual fault detection through the spatial and temporal correlation between sensor data to ensure the accuracy and reliability of road object detection.

The main contributions of this work are summarized as follows.

(1) Compared with the previous object detection network, we use a more lightweight feature pyramid network (FPN) [25] structure to ensure the real-time performance of the data fusion system when processing large-scale multi-modal data.
(2) The proposed FDA mechanism in the data fusion framework guarantees the elimination of the sensor fault signal in real time, and the accuracy and reliability of the detection results.
(3) Finally, the environmental perception data captured in multiple scenes are used to conduct experiments to verify the real-time, accuracy and reliability of the fusion framework from the aspects of 2D object detection performance and fault avoidance performance.

2 Data Fusion Framework

Most data fusion methods mainly study the fusion scheme and complete the corresponding object detection tasks. However, sensor faults are not considered in these studies. In this paper, an environmental perception scheme based on multiple cameras is proposed, in which the framework can use any type and number of cameras if the external parameters are known. The fields of view of multiple cameras are interleaved to form a redundancy, and the fusion framework can diagnose the credibility of each sensor data in real time through the correlation between the space and time of the camera data. Furthermore, to ensure the real-time performance of object detection when processing large-scale multi-modal data, this paper proposes a lightweight design for the YOLO V3 network, which greatly reduces the load on the system calculation and storage unit.

Analogous to Adaboost algorithm, this paper treats each camera as a weak classifier. The data fusion algorithm combines these weak classifiers to form a more accurate strong classifier, which is used to detect the position, attitude and rotation of target objects from the environmental information. The fusion results will be fed back to the data fusion framework to correct the weight of each sensor in real-time. The overall structure of the fusion framework is shown in Figure 1.

The framework is mainly composed of three parts: The first part is the RPN, which fuses the image information collected by each camera sensor and outputs a series of region proposals that may contain targets. The second part is the FDA, which is designed to improve the fault-tolerance of the system. The data fusion framework calculates the local and global confidence of the sensor data of the current frame in real time, and removes the noise signal during the fusion process, thereby ensuring the reliability of data fusion. The global non-maximum suppression (GNMS) processes the prediction results D_i of RPN and the global confidence K_i of the sensor generated by FDA, and the detection results D_{fusion} are obtained after data fusion.

3 Design of Region Proposal Networks

Image data contains rich color and background information, which is convenient for deep networks to separate foreground and background. Inspired by Ref. [6], this paper uses the anchor area as the basic unit of 2D object detection, on which it regresses the prediction of the target position and type. Meanwhile according to the spatial correlation of the image data, the deep network infers the confidence coefficient C for each local area of the image, which is used to subsequently eliminate redundant and erroneous data.

3.1 Feature Extraction Network (FEN)

After acquiring a frame of image, it is necessary to compress and encode it, and extract the topology information between pixels through a neural network. The more complex the level of the FEN, the larger the perception domain of the output feature map, however it also loses more local information. To reduce network parameters, this paper uses residual neural network [3] as the basic structure of the FEN. When designing a network framework, the size of feature maps is reduced by introducing priori features, thereby the computing power required by the network is greatly reduced. The backbone structure of the network is shown in Table 1.

The load of the calculation and storage unit of the neural network can be estimated by the network computing power. For a CNN layer, its computing power can be calculated by Eq. (1):

$$FLOPs = 2HW(C_{in}k^2 + 1)C_{out}, \tag{1}$$

where H and W represent the height and width of the feature map, and C_{in}, C_{out} are the number of input/output channel of the feature map, k represents the kernel size. In the FEN, the two-layer 3 × 3 convolutional layer is first used to reduce the size of the image, which requires the computing power of 3.47×10^7 FLOPs. Then the subsequent residual structures are brought into Eq. (1) for calculation and addition, and the total computing power of the network FE layer is 0.686×10^9 FLOPs, compared to the 18.569×10^9 FLOPs computing power required by DarkNet-53, the network computing power required in this paper is its 1/27.

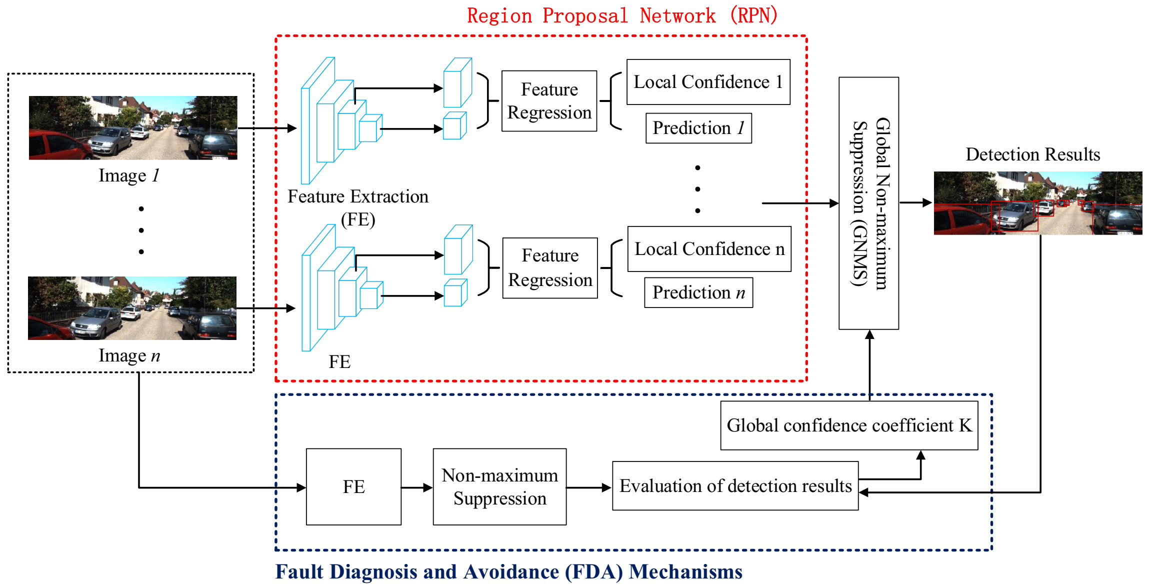

Figure 1 Data fusion framework

Table 1 Feature extraction network structure

	Type	Channel	Filter size	Output
	Input	3		416 × 416
	Convolutional	16	3 × 3/4	104 × 104
	Convolutional	32	3 × 3/2	52 × 52
2×	Convolutional	32	3 × 3	
	Convolutional	32	3 × 3	
	Residual			52 × 52
	Convolutional	64	3 × 3/2	26 × 26
2×	Convolutional	64	3 × 3	
	Convolutional	64	3 × 3	
	Residual			26 × 26
	Convolutional	128	3 × 3/2	13 × 13
2×	Convolutional	128	3 × 3	
	Convolutional	128	3 × 3	
	Residual			13 × 13
	Required computing power: 0.686 × 10^9 *FLOPs*			

The lightweight FPN network extracts 26 × 26 and 13 × 13 feature map in backbone to construct a feature pyramid, and calculates the output vector $\{x, y, w, h, C, P(0), P(1), P(2)\}$ including local confidence C for each local area of the image, the bounding box regression variables $\{x, y, w, h\}$ and the target category prediction probability $\{P(0), P(1), P(2)\}$. Note that the target area obtained by the ith camera after passing through the lightweight FPN network is predicted to be D_i. The GNMS module specified later will synthesize each camera sensor to predict D_i and its corresponding real-time confidence K_i, the RPN unit finally infers the location of the target D_{fusion} in the 2D image and transmits it to the bounding box regression network in the data fusion framework.

3.2 Estimate RoI Area

The FEN compresses and encodes the original image to form multiple feature maps with different degrees of compression, similar to the feature pyramid in the SIFT operator. In this paper, the feature maps after compression and encoding at different scales are divided into anchor regions, and the target position and local confidence information are extracted. Specifically, for a *grid* × *grid* feature map, three fixed length and width anchors are placed at each position of the feature map to detect the target object. The anchor is essentially a two-dimensional rectangular bounding box, which is generally represented by a quaternion vector $\{x, y, w, h\}$ that is composed of its position information and length and width information. The network also predicts the probability that its internal target belongs to each category $\{P(0), P(1), \ldots, P(n)\}$ for each anchor. The network detects three kind of targets (pedestrians, vehicles, and bicycles), thus the output vector size of each object detection layer is *batch_size* × *anchor_num* × *grid* × *grid* × 8$\{x, y, w, h, C, P(0), P(1), P(2)\}$.

The initial size of Anchor is obtained by k-means clustering on the KITTI training set [26]. When the size of Anchor is closer to the size of the target itself, the network is more likely to locate the target correctly under the local area divided by Anchor. This paper designs two feature layers for the lightweight FPN network, so when clustering $k = 6$, the final sizes of the 6 anchors obtained by clustering are (40, 37), (96, 56), (79, 163), (170, 93), (256, 155), (372, 210).

3.3 Loss function

After the construction of the network structure is completed, it is necessary to use the loss function to guide the convergence path of the network parameters to induce the network to approach the set nonlinear function in continuous self-learning. The network loss function is mainly composed of three parts, including bounding box regression loss l_{bbox}, bounding box object detection loss l_{conf}, and target classification loss l_{cls}, which correspond to the output vector of the model respectively. The confidence of the bounding box and the predicted value of the target category are output in the form of probabilities, thus l_{conf}, l_{cls} are calculated using the cross-entropy loss function. However, the output of $\{x, y, w, h\}$ is a specific value, and they are taken into the square error loss function model for calculating l_{bbox}. The specific expressions of the three loss functions are designed as follows:

$$
\begin{aligned}
l_{conf} &= \sum_{i=0}^{grid^2} \sum_{j=0}^{anch} \left(\lambda_{noobj} \mathbf{1}_{ij}^{noobj} - \mathbf{1}_{ij}^{obj} \right) \log C_{ij}, \\
l_{bbox} &= \sum_{i=0}^{grid^2} \sum_{j=0}^{anch} \mathbf{1}_{ij}^{obj} \Big[(x_{ij} - \hat{x}_{ij})^2 + (y_{ij} - \hat{y}_{ij})^2 \\
&\quad + (\sqrt{\omega_{ij}} - \sqrt{\hat{\omega}_{ij}})^2 + (\sqrt{h_{ij}} - \sqrt{\hat{h}_{ij}})^2 \Big], \\
l_{cls} &= \sum_{i=0}^{grid^2} \sum_{j=0}^{anch} \mathbf{1}_{ij}^{obj} \sum_{c \in classes} \left[P_{ij}(c) - \hat{P}_{ij}(c) \right]^2,
\end{aligned}
\tag{2}
$$

where λ_{ij}^{noobj} represents the weighting factor that does not include the target Anchor, *grid* represents the size of the output feature map, $\mathbf{1}_{ij}^{obj}$ represents whether the Anchor (i, j) is responsible for the target, *anch* represents the number of anchors in each unit.

During network training, each pixel in the output feature map will select the anchor that is the largest Intersection-over-Union (IoU) with the target object. These anchors corresponding to $\mathbf{1}_{ij}^{obj}$ is set to 1, and their predicted value will participate in the calculation of l_{bbox} and

l_{cls}. For anchors whose $\mathbf{1}_{ij}^{obj}$ is 0, if their IoU with any target is less than 0.5, it is considered that there is no object in the Anchor, and the corresponding $\mathbf{1}_{ij}^{noobj}$ is set to 1, and its predicted confidence C will participate in the calculation of l_{conf} as a penalty term in logarithmic form.

In general, the anchor with no target in the output accounts for the vast majority, which will make the value of $\sum_{i=0}^{grid^2}\sum_{j=0}^{A}1_{ij}^{noobj}\log C_{ij}$ directly affect the calculation of l_{conf}. The network tends to predict all C_{ij} to a very small value, such a loss function is meaningless. For this purpose, the distribution of the input data needs to be artificially modified by the weight coefficient $\mathbf{1}_{ij}^{noobj}$. Similarly, compared with traditional object detection tasks, pedestrian detection for autonomous driving pays more attention to the accuracy of target position prediction. The single sensor object detection network is used as the "preprocessing" link of data fusion, and its estimation of the target position will directly affect the data fusion effect. Therefore, when calculating the total loss, multiply l_{bbox} by a larger weighting factor λ_{bbox} to increase the "penalty" of the regression error of the bounding box to the network. The final network loss function is calculated by

$$l = l_{conf} + \lambda_{bbox}l_{bbox} + l_{cls}. \tag{3}$$

4 Fault Diagnosis and Avoidance Mechanism

During the driving process of the autonomous vehicle, severe weather conditions such as haze and rain may be encountered, or the camera lens may be blocked by dirt, or the data transmission bus may be cut off. These problems will introduce noise into the environmental perception data, leading to problems, such as missed detection and wrong detection of targets, which bring huge safety hazards to autonomous driving. A fault diagnosis and avoidance mechanism of perception system is designed to ensure the redundancy of sensor data. Through the redundant information for fault diagnosis, the accuracy of the sensor data at the current moment can be judged and the fault signal can be eliminated. In this paper, the FDA mechanism is realized by setting up dynamic weights. The data fusion framework calculates the global confidence level K and the local confidence level C for the sensor data in real time, which are used to deal with the local faults and global faults issues. According to the confidence coefficient (K, C), the GNMS algorithm eliminates redundant and faulty signals during the fusion of the RoI region to obtain accurate fusion results as the output of object detection.

4.1 Fault Analysis of Environmental Perception Sensor

For the camera, its imaging structure makes the image data susceptible to the light in the environment. Poor lighting, complex background information, and changes of view will have a certain impact on the perception accuracy of the sensors. The noise caused by the environment on the camera sensor can be roughly divided into two types:

- **Global fault**. Noise signals will be distributed throughout the image, such as bad environment, complex background, and camera out of focus.
- **Local fault**. Parts of the photosensitive chip are not working, such as data truncation, partial occlusion and lens stains.

During the driving process of the autonomous vehicle, the view angle of target, background and lighting conditions are constantly changing. Our fault diagnosis mechanism uses a dynamic weight method to update the weight in real time according to each frame of image data to ensure the reliability of the environmental perception system. For global sensor faults, we combine the Kalman filter algorithm to estimate the accuracy of each sensor data and its contribution to the final result in real time, and sequentially modify the overall confidence coefficient of the sensor. For local faults, RPN are used to calculate confidence coefficient for each local area of the image in real time. The data fusion framework uses GNMS to combine the local coefficients and global coefficients to ensure the accuracy and reliability of the data fusion results.

4.2 Global Confidence Coefficients Correction

In an autonomous driving system, the measurement areas of multiple sensors intersect each other, and there is redundancy in the acquired environmental perception data. The data fusion network fuses the sensing data according to the global and local confidence coefficients of the sensors to obtain the theoretically optimal object detection results. These results will be fed back to the fusion network as Ground Truth and used to correct the global confidence coefficient K of each sensor. This paper used the Kuhn–Munkres (KM) algorithm and Kalman filter to assist the correction of global confidence coefficient K, where the KM algorithm is used to solve the maximum weight matching problem in the bipartite graph. In this paper, it is known that the tracking mark T predicted in the previous frame and the object detection result D in the current frame are known, and they respectively constitute two subsets of the bipartite graph $\mathcal{G}$. The KM algorithm finds the optimal connection M from $\mathcal{G}$ according to certain rules and guarantees:

$$\sum_{i=0}^{m-1} M_{ij} \leq 1, \sum_{i=0}^{n-1} M_{ij} \leq 1, \tag{4}$$

that is, each test result can only be matched to one mark. Before running the KM algorithm, it is necessary to obtain the correlation coefficient $G(i, j)$ between the points in the bipartite graph, which represents the strength of the similarity between the marker T_i and the detection result D_i. In this paper, the IoU between the bounding boxes is used as the correlation coefficient. The intersection ratio comprehensively reflects the similarity of the position and size of the two bounding boxes, and is a simple and effective solution for calculating the correlation coefficient. After calculating the correlation coefficient matrix, we exclude the combination of $G(i, j) < 0.2$. The two bounding boxes in these combinations are too far apart and cannot belong to the same object. The correlation coefficient corresponding to the excluded combination is set to 0, indicating that there is no connection between the two points. The KM algorithm traverses the remaining connections to find the optimal combination M, so that the sum of the correlation coefficients between Tracker and Detection is maximized under this combination, namely:

$$\hat{M} = \arg\max_{M} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} M_{ij} G_{ij}. \tag{5}$$

After obtaining the matching matrix M between Tracker and Detection, we use Kalman filter to update the object tracking information in Tracker. The states of vehicles and pedestrians on the road are described by the position of the target in the world coordinate system p and the current speed v, where $p = [\,x\ y\ z]^{\mathrm{T}}$, $v = [\,v_x\ v_y\ v_z]^{\mathrm{T}}$. Since the real state of the target cannot be obtained, Kalman filter assumes that the target obeys a Gaussian distribution with a mean value of s_k and a variance of Σ_k in the kth frame. When the sampling frequency of the environmental perception sensor is high enough, the motion pattern of the target between two adjacent frames can be approximately regarded as a uniform linear motion, and the equation for updating the target state can be written as follows:

$$\begin{aligned} \hat{s} &= \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} s_{k-1} = F s_{k-1}, \\ \hat{\Sigma}_k &= F \Sigma_{k-1} F^{\mathrm{T}}. \end{aligned} \tag{6}$$

However, the law of target motion cannot be a completely uniform linear motion. As the time difference Δt increases, the error between the predicted value and the real value will also accumulate. Therefore, it is necessary to correct the predicted value in real time through the measurement data fed back from the sensor. In this paper, the sensor data is fused to obtain the position, size and steering information of the target in the 3D reference frame. The speed information measured by the sensor is obtained by dividing the difference between the detection and the tracker position by the time. Combine the position and velocity information measured by the sensor and record it as z_k, and fuse the target state prediction distribution P_{pred} and the sensor measurement distribution P_{sensor} through Kalman filter, we get:

$$\begin{aligned} H s_k &= H \hat{s}_k - L(z_k - H \hat{s}_k), \\ H \Sigma_k H^{\mathrm{T}} &= H \hat{\Sigma}_k H^{\mathrm{T}} - L H \hat{\Sigma}_k H^{\mathrm{T}}, \\ L &= H \hat{\Sigma}_k H^{\mathrm{T}} (H \hat{\Sigma}_k H^{\mathrm{T}} + \Sigma_k')^{-1}. \end{aligned} \tag{7}$$

Finally, the reconciled distribution is projected back to the original target state space, we have

$$\begin{aligned} s_k &= \hat{s}_k - L'(z_k - H \hat{s}_k), \\ \Sigma_k &= \hat{\Sigma}_k - L' H \hat{\Sigma}_k, \\ L' &= \hat{\Sigma}_k H^{\mathrm{T}} (H \hat{\Sigma}_k H^{\mathrm{T}} + \Sigma_k')^{-1}. \end{aligned} \tag{8}$$

From Eqs. (8) and (10), the target state prediction distribution $P_k = \mathcal{N}(s_k, \Sigma_k)$ is obtained by Kalman filter. This distribution will be recorded in the corresponding Tracker for prediction of the target state of the frame $k + 1$. The location information in s_k will be used to correct the global weights (K_0, K_1, K_2, K_3) of the four cameras. This operation is completed by the following steps:

(1) For the information D_i of the target bounding box predicted by the 2D object detection network, the traditional NMS method is used to generate individual prediction information for each camera sensor $\hat{D}_i$.

(2) Extract the target location information p_k in s_k, and map it to the pixel coordinate system of the camera$_0$ according to the perspective projection change, and set it as c_k.

(3) Use all predictions c_k in frame k as Ground Truth to calculate the mean IoU value M_i of $\hat{D}_i$ under c_k. Specifically, for each prediction bounding box $\hat{D}_i[j]$ in $\hat{D}_i$, find the nearest c_k to its center to match. Construct a bounding box with the size of $\hat{D}_i[j]$ as the corresponding c_k, and calculate its IoU value with $\hat{D}_i[j]$. All IoU values are weighted and averaged according to the confidence coefficient C of $\hat{D}_i[j]$, and the mean IoU value of $\hat{D}_i$ under c_k is obtained.

(4) Update the overall confidence of each sensor according to the mean IoU. For the sensor i, the confidence coefficient K_i can be updated as follows:

$$\begin{aligned} \alpha &= \frac{1}{2}\ln\left(\frac{mean(IoU)}{1-mean(IoU)}\right), \\ K_i' &= K_i \exp\left[\alpha(IoU_i - mean(IoU))\right]. \end{aligned} \tag{9}$$

4.3 Global Non-maximum Suppression

At the moment of $t-1$, the global confidence level of the sensor K_i is updated by Eq. (11), and it will be fed back to the GNMS module for the fusion of the next frame of data. In the next stage, the image data at time t is input into the data fusion network, and after processing by the lightweight FPN network, a ROI area set $\{D_i\}_{i=1}^{N}$ is generated. The GNMS algorithm performs fault diagnosis according to the confidence coefficient (K, C) of the current frame, redundant and wrong sensor data are eliminated, and accurate fusion results are generated.

Compared with traditional environment perception algorithms, GNMS can filter sensor data based on real-time calculated global and local confidence coefficients, and complete data fusion tasks in a targeted manner. After the image ROI set $\{D_i\}_{i=1}^{N}$ is suppressed by the global non-maximum value, the redundant and faulty data is eliminated, and the missing data is completed. Finally, the results of object detection after sensor fusion is obtained.

5 Evaluation of Network Performance

5.1 Experimental Setup

The results in the following sections are all based on the settings in this section. The model is first pre-trained on the COCO [27] dataset for 500000 steps with a batch size 8. Training by Adam optimizer follows the learning rate which is warmed up to 10^{-3} in the first 1000 steps and is multiplied by a hyper-parameter $l\_drop = 0.1$ at step 400000 and step 450000. Then the pre-trained model is trained on the KITTI dataset for 100 epochs with a fixed learning rate. The weighting factor of bounding box regression loss mentioned in Section 3 is set to 1 for faster convergence. All training experiments are executed on two RTX 2080Ti GPUs.

5.2 Performance of Object Detection

The KITTI dataset provides researchers with a completed model evaluation program. The program improves the average precision (AP) calculation method proposed in PASCAL VOC. It does not care about targets that are too small in the environment and targets that are too far away from the camera. To a certain extent, the requirements on the model are relaxed. At the same time, KITTI's model evaluation program divides the task into three difficulty levels according to the size of the target bounding box and the degree of occlusion: Easy, Moderate and Hard. The specific division of each difficulty is shown in Table 2.

Table 2 Difficulty division of KITTI dataset

Difficulty level	Minimum box height	Maximum occlusion	Maximum truncation
Easy	40 pixels	Fully visible	15%
Moderate	25 pixels	Partial occlusion	30%
Hard	25 pixels	Hard to identify	50%

Table 3 Performance comparison of 2D object detection

Network	Running time (s)	Easy (%)	Moderate (%)	Hard (%)
Faster-RCNN	2	88.97	83.16	72.62
RetinaNet [28]	0.2	93.97	82.73	68.37
RefineNet [29]	0.2	91.91	81.01	65.67
MonoFENet [30]	0.15	91.68	84.63	76.71
ResNet-RRC	0.06	91.45	85.33	74.24
Fast-SSD	0.06	85.19	66.79	57.89
YOLO800	0.13	78.93	74.31	63.83
The new method	0.02	95.36	88.20	86.44

The evaluation procedure of the KITTI dataset divides the objects into three categories: vehicles, pedestrians, and bicycles. For vehicle targets, use AP_{70} to evaluate network performance, and for pedestrian and bicycle targets, calculate AP_{50}. Since the KITTI does not provide the official label of the test set, we adopt the set aside method for network performance evaluation, and randomly selected 1000 images from the training set as the test set. Note that these 1000 images do not participate in the network training. Finally, we compile the evaluation results of some well-known networks on the Leader Board of KITTI's official website to compare with our network performance, as shown in Table 3.

The precision and recall (PR) curve drawn by the network in this paper under Easy, Moderate and Hard difficulty is shown in Figure 2(a). At the same time, the PR_{50} curve obtained under all test sets compares YOLO_V3, as shown in Figure 2(b). The object detection network needs to run on four 2D images simultaneously. Considering the GPU memory and the real-time performance of the algorithm, we have adopted a series of methods to reduce network parameters and required computing power, such as HOG feature extraction, residual network. Based on

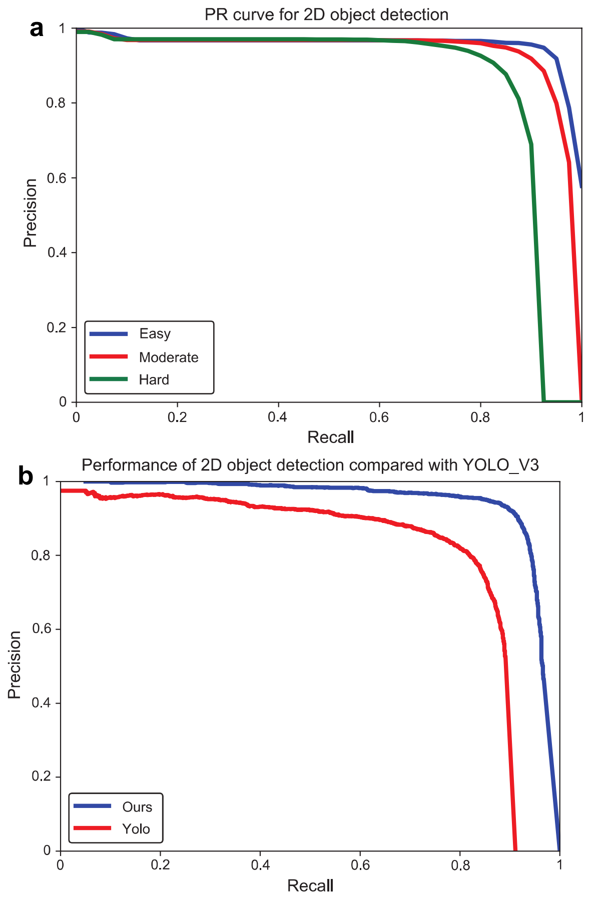

Figure 2 Performance of 2D object detection

Table 3 and Figure 2, although our object detection network has been greatly optimized in terms of parameters and computing speed, it still guarantees considerable accuracy and recall rate, and its object detection performance can fully meet the current requirements of autonomous vehicles for environmental perception accuracy.

5.3 Fault Diagnosis and Avoidance Performance

The FDA framework of this paper is mainly designed for two types of faults, namely, the global and local faults. In the experiment, the camera out-of-focus problem is used to simulate the global faults. By down-sampling the image, the resolution of the entire image is reduced, and the image becomes blurred, and the target object in the image could not be identified and difficult to locate. Assuming that the original image matrix is *Img* and the output fault image matrix is *Err*, for the camera out-of-focus fault, the down-sampling operation as shown in Eq. (10) can be used to simulate:

Table 4 Comparison of fault tolerance performance

Video number	Out-of-focus (FDA)	Out-of-focus	Truncated (FDA)	Truncated
09-26-0009	0.8091	0.7390	0.8115	0.6858
09-26-0017	0.8820	0.8014	0.8881	0.6661
09-26-0051	0.7710	0.6542	0.7708	0.5901
09-26-0056	0.8812	0.7556	0.8816	0.7903
09-26-0057	0.8364	0.7461	0.8294	0.7334

$$Err(i,j) = Img(i - (i \bmod s), j - (j \bmod s)), \quad (10)$$

where s is the step size of the down-sampling operation.

For local faults, the problem of image truncation is used to simulate, and the row arrangement of the local area of the image is disordered to simulate the more serious garbled effect. Because the fault signal is generated on the basis of the original data, this kind of garbled code also makes the network more prone to error detection problems, and the comparison effect is more obvious in the experiment. Assumed that the *shuffle*() function can slice and shuffle a series of continuous data, the way to simulate local garbled characters can be expressed by

$$Err(i,j) = Img(shuffle(i), j). \quad (11)$$

As a control group, the network uses exactly the same network parameters, but removes the GNMS module and Kalman filter part, and only uses the weight-based method for simple fusion. After simulating the sensor fault on the KITTI Raw-Data dataset through the software, we use our fusion framework and the control group to run on it to obtain object detection results, and calculate the Mean-IoU value obtained by the two types of fault data when processing out-of-focus and truncation. The results are shown in the Table 4.

Draw the mean IoU curve of the network under the video data with the number of frames as the x-axis and the IoU value of each frame as the y-axis, which can more intuitively compare the FDA performance of the two networks. Select the video data numbered 09-26-0056 as input, and draw the mean IoU curve of this network and the traditional data fusion network under sensor fault as shown in Figure 3.

To detect the maximum fault ability of the data fusion framework designed in this paper, we impose more serious global and local sensor faults on the sensor data of multiple cameras, and measure the mean IoU value according to the same method. When two cameras have local and global faults simultaneously, the mean IoU value obtained by our network and the control group is shown in the Table 5.

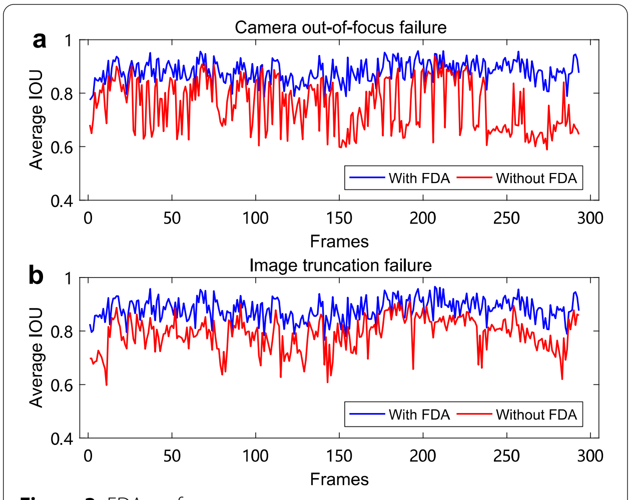

Figure 3 FDA performance

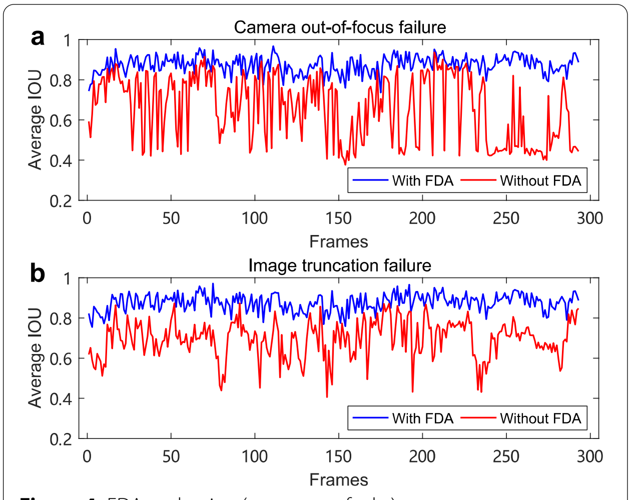

Figure 4 FDA evaluation (two sensor faults)

Table 5 Performance comparison of multi-sensor fault tolerance

Video number	Out-of-focus (FDA)	Out-of-focus	Truncated (FDA)	Truncated
09-26-0009	0.8040	0.6938	0.7977	0.5890
09-26-0017	0.8540	0.7622	0.8603	0.5046
09-26-0051	0.7619	0.5649	0.7751	0.6319
09-26-0056	0.8764	0.7288	0.8759	0.6959
09-26-0057	0.8117	0.6849	0.8282	0.6659

Select the video data numbered 09-26-0056 as input, and draw the IoU curve obtained when two cameras fail at the same time, as shown in Figure 4.

When the same fault is applied to three of the four cameras, the fault avoidance system of the network in this paper fails, and the mean IoU curve has dropped significantly, as shown in Figure 5(a).

Analyzing the reason for the fault of the network is that our network uses each detection result as ground truth feedback and corrects the sensor weight. When the same fault is applied to the three cameras, these three sets of detection results will obtain similar fault detection results under the influence of the same mode of fault, which makes the initial fusion detection result and the real ground truth have a large deviation, and makes the fault tolerance system invalid. When garbled characters with different positions and different generation modes are applied to the three cameras, and the mean IoU curve obtained on such data is shown in Figure 5(b), and the faults imposed on the sensor are correctly eliminated.

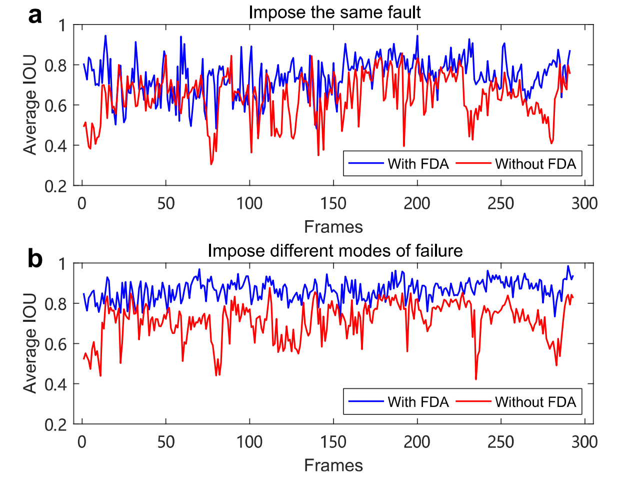

Figure 5 FDA evaluation (three sensor faults)

Printing the detection results of the data fusion framework in this paper and the control group on the original image data can more intuitively show the impact of fault signals on object detection accuracy and the performance of our network to eliminate fault detection results. The comparison of the detection results of the two networks under the interference of the fault signal is shown in Figure 6.

It can be seen from the above experiments that when two or more cameras have real data, the fault diagnosis mechanism designed in this paper can effectively avoid the noise information caused by the global and local faults of the sensor. Even if the sensor does not have a serious fault phenomenon, the global and local confidence methods can also eliminate the perception errors during network operation in time, and make full

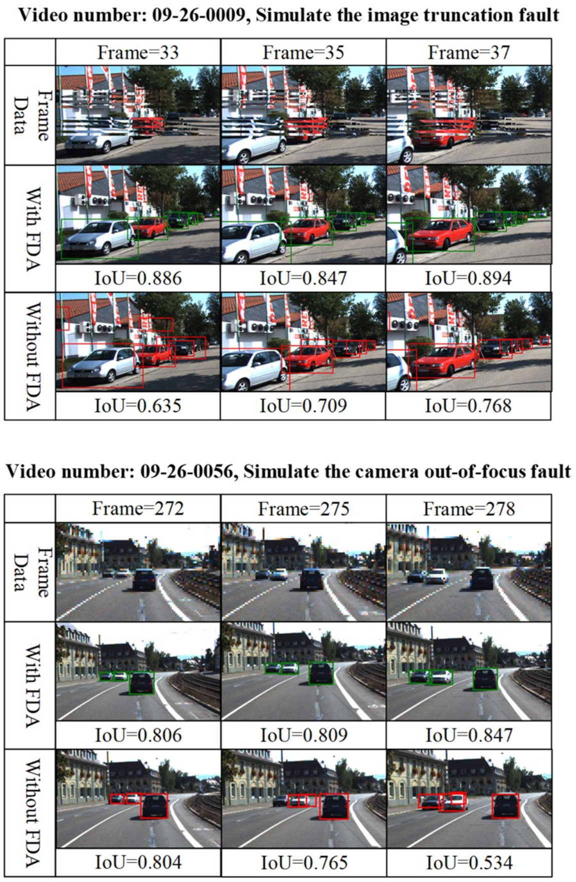

Figure 6 Comparison of fault tolerance performance

use of more accurate sensor data to obtain the best data fusion effect.

6 Conclusions

(1) Aiming at the problem of object detection in autonomous driving scenarios, a data fusion framework with fault diagnosis mechanism is proposed, which realizes the mutual diagnosis of faults between environmental perception sensors by setting global and local confidence levels, and eliminates the noise information introduced when the sensors collect environmental data.

(2) The data fusion network in this paper greatly reduces the load on the computing and storage units of the multi-source environment sensing system while ensuring the accuracy of object detection through the experiments on KITTI dataset.

(3) The proposed FDA mechanism enables the data fusion network to timely eliminate fault information when the sensor has a serious global or local fault, ensuring the accuracy and reliability of data fusion.

Acknowledgements
Not applicable.

Authors' contributions
HG was in charge of the whole trial; HP wrote the manuscript; QS perfected the formulas and programmed the data; WS gave some advices on the manuscript. All authors read and approved the final manuscript.

Authors' Information
Huihui Pan, received the Ph.D. degree in control science and engineering from *Harbin Institute of Technology, China*, in 2017, and the Ph.D. degree in mechanical engineering from the *Hong Kong Polytechnic University*, Hong Kong, China, in 2018. He is currently an associate Professor with the *Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin, China*. His current research interests include nonlinear control, vehicle dynamics, and intelligent vehicles.

Weichao Sun, received the Ph.D. degree in control science and engineering from *Harbin Institute of Technology, China*, in 2013. He is currently a full Professor with the *Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, China*. His current research interests include intelligent vehicles, motion control and robotics.

Qiming Sun, received the M.S. in control science and engineering from *Harbin Institute of Technology, China*, in 2020. Since September 2020, he has been with the *HUAWEI Technologies Co., Ltd., Beijing, China*.

Huijun Gao (IEEE Fellow), received the Ph.D. degree in control science and engineering from *Harbin Institute of Technology, China*, in 2005. From 2005 to 2007, he carried out his post-doctoral research with the *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada*. Since 2004, he has been with the *Harbin Institute of Technology*, where he is currently a Full Professor, and the Director of the *Research Institute of Intelligent Control and Systems*. His research interests include intelligent and robust control, robotics, mechatronics, and their engineering applications. Dr. Gao is an Vice President of *IEEE Industrial Electronics Society*, a Council Member of the *International Federation of Automatic Control (IFAC)*, and the distinguished lecturer of *IEEE Systems, Man and Cybernetics Society*. He also serves as the Co-Editor-in-Chief for the *IEEE Transactions on Industrial Electronics*, a Senior Editor for the *IEEE/ASME Transactions on Mechatronics*, and an Associate Editor for *Automatica*, the *IEEE Transactions on Cybernetics*, and the *IEEE Transactions on Industrial Informatics*.

Funding
Supported by the National Natural Science Foundation of China (Grant U1964201, Grant 61790562 and Grant 61803120), and by the Fundamental Research Fundsfor the Central Universities.

Competing Interests
The authors declare no competing financial interests.

Author Details
[1]Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China. [2]The Ningbo Institute of Intelligent Equipment Technology Co., Ltd, Ningbo 315200, China.

Received: 28 November 2020 Revised: 3 April 2021 Accepted: 10 May 2021
Published online: 20 July 2021

References

[1] G Xiang, X Wang, D Liang. Background reconstruction and object extraction based on color and object tracking. *Chinese Journal of Mechanical Engineering*, 2006, 19(3): 471-474 .

[2] A Krizhevsky, I Sutskever, G E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012: 1097-1105.
[3] K He, X Zhang, S Ren, et al. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 770-778.
[4] R Girshick, J Donahue, T Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014: 580-587.
[5] R Girshick. Fast r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*, 2015: 1440-1448.
[6] S Ren, K He, R Girshick, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 2015: 91-99.
[7] V Murugan, V Vijaykumar, A Nidhila. A deep learning rcnn approach for vehicle recognition in traffic surveillance system. *2019 International Conference on Communication and Signal Processing (ICCSP)*, 2019: 0157-0160.
[8] X Liu, W Liu, T Mei, et al. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. *European Conference on Computer Vision*, 2016: 869-884.
[9] K Shi, H Bao, N Ma. Forward vehicle detection based on incremental learning and fast r-cnn. *2017 13th International Conference on Computational Intelligence and Security (CIS)*, 2017: 73-76.
[10] Y Qu, L Jiang, X Guo. Moving vehicle detection with convolutional networks in uav videos. *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, 2016: 225-229.
[11] J Redmon, S Divvala, R Girshick, et al. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 779-788.
[12] J Redmon, A Farhadi. Yolo9000: better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 7263-7271.
[13] J Redmon, A Farhadi. Yolov3: An incremental improvement. *arXiv preprint* arXiv:1804.02767 *(2018).*
[14] W Liu, D Anguelov, D Erhan, et al. Ssd: Single shot multibox detector. *European Conference on Computer Vision*, 2016: 21-37.
[15] N Chodosh, C Wang, S Lucey. Deep convolutional compressed sensing for lidar depth completion. *Asian Conference on Computer Vision*, 2018: 499-513.
[16] F Mal, S Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018: 1-8.
[17] A Eldesokey, F Felsberg, F S Khan. Propagating confidences through cnns for sparse data regression. *arXiv preprint* arXiv:1805.11913 *(2018).*
[18] S Shao, W Sun, R Yan, et al. A deep learning approach for fault diagnosis of induction motors in manufacturing. *Chinese Journal of Mechanical Engineering*, 2017, 30(6): 1347-1356.
[19] H Cho, T W Seo, B V Kumar, et al. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014: 1836-1843.
[20] X Chen, H Ma, J Wan, et al. Multi-view 3D object detection network for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 1907-1915.
[21] J Ku, M Mozifian, J Lee, et al. Joint 3D proposal generation and object detection from view aggregation. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: 1-8.
[22] Y He, M Ji, C Zhang, et al. Global exponential stability of neural networks with time-varying delay based on free-matrix-based integral inequality. *Neural Networks*, 2016, 77: 80-86.
[23] F Shen, Y Hui, R Yan, et al. A new penalty domain selection machine enabled transfer learning for gearbox fault recognition. *IEEE Transactions on Industrial Electronics*, 2020, 67: 8743-8754.
[24] S Shao, S McAleer, R Yan, et al. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 2018, 15(4): 2446-2455.
[25] T Lin, P Dollár, R Girshick, et al. Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 2117-2125.
[26] J Fritsch, T Kuehnl, A Geiger. A new performance measure and evaluation benchmark for road detection algorithms. *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
[27] T Y Lin, M Maire, S Belongie, et al. Microsoft COCO: Common objects in context. *European Conference on Computer Vision*, Springer International Publishing, 2014: 740-755.
[28] T Lin, P Goyal, R Girshick, et al. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017: 2980-2988.
[29] R N Rajaram, E Ohn-Bar, M M Trivedi. Refinenet: Refining object detectors for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2016, 1(4): 358-368.
[30] W Bao, B Xu, Z Chen. MonoFENet: Monocular 3D object detection with feature enhancement networks. *IEEE Transactions on Image Processing*, 2020, 29: 2753-2765.