IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 21, 2020, accepted March 3, 2020, date of publication March 9, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979260

A Traffic Surveillance Multi-Scale Vehicle Detection Object Method Base on Encoder-Decoder

FENG HONG[1,2], CHANG-HUA LU[1], CHUN LIU [1], RU-RU LIU[2], AND JU WEI[3]

[1]School of Computer and Information, Hefei University of Technology, Hefei 230026, China
[2]School of Mechanical and Electrical Engineering, Chizhou University, Chizhou 247100, China
[3]School of Internet, Anhui University, Hefei 231100, China

Corresponding authors: Feng Hong (hongfeng_tougao@163.com) and Chun Liu (liuchuntougao@163.com)

This work was supported in part by the Major National Science and Technology Projects, China, under Grant JZ2015KJZZ0254, and in part by the National High Technology Research Development Plan (863), China, under Grant 2014AA06A503.

ABSTRACT Aiming at the problem that it is difficult for traffic monitoring videos to detect multi-scale vehicle targets, especially small vehicle targets in complex scenarios, a codec-based vehicle detection algorithm is proposed. This algorithm is based on YOLOv3. In order to solve the multi-scale vehicle target detection problem, a new multi-level feature pyramid structure added with the codec module to detect vehicle targets of different scales. The experimental results on the KITTI dataset and UA-DETRAC dataset confirm that the algorithm in this paper has achieved good detection results for vehicle targets in various environments and at various scales in the surveillance video, especially for small vehicle targets, which can better meet the actual application demand.

INDEX TERMS Surveillance video, vehicle detection, codec, convolutional neural network.

I. INTRODUCTION

Vehicle target detection in video surveillance is an important subject in intelligent transportation systems. By accurately detecting the vehicles in the surveillance video, follow-up research work such as license plate recognition, vehicle tracking, and traffic flow statistics can be further performed. Therefore, the vehicle detection method is the premise and basis of these follow-up research work, and has high practical application value.

Vehicle target detection algorithms can be divided into traditional vehicle target detection algorithms and deep learning-based vehicle target detection algorithms. Traditional vehicle target detection is mainly to manually extract features and then use a classifier to determine whether the area belongs to a real vehicle. For example, FELZENSZWALB *et al.* [1] proposed a multi-scale deformable parts model (DPM) [2] to perform object detection including automobiles. Dalal and Triggs *et al.* proposed to use directional gradient histogram (Histogram of Gradient, HOG) features [3], combined with a linear support vector machine (SVM) for detection. This type of method has the advantage of high speed, but it has low detection accuracy in complex environments (such as partial occlusion, shape differences, scale changes, insufficient night light, and poor visibility in bad weather).

In recent years, with the development of artificial intelligence and the increase of computing speed in recent years, deep learning methods have achieved outstanding results in computer vision, speech recognition, text recognition and other fields [4]–[10]. The publication of introduced the convolutional neural network AlexNet into the target detection task [11]–[13], which opened the prelude to the deep learning-based target detection algorithm. There are two types of frameworks for object detection algorithms based on deep learning: two-stage frameworks and one-stage frameworks. For the two-stage detection framework: In 2012, GIR-SHICK R proposed the first CNNs-based target detection framework RCNN [14]. The algorithm's Mean Average Precision (MAP) is 30% higher than the traditional algorithm. In 2014, He *et al.* proposed the SPP net [15], which uses a spatial pyramid pooling layer to reduce the size limit of convolutional neural networks. Subsequently, a series of excellent two-stage detectors have emerged, such as Fast-RCNN [16], Faster R-CNN [16], Mask R-CNN [17], etc. The accuracy of the two-stage detection framework is very high, but Slow

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

FIGURE 1. Vehicles with large imaging scale spans easily cause missed detection (yellow boxes are detected targets and red boxes are missed targets).

speeds are common. For the 1-stage detection framework: Joseph *et al.* proposed You Only Look Once (YOLO) [18] in 2016, and used the detection process as the regression task for the first time. Later Liu *et al.* proposed a Single Shot Multi-Box Detector (SSD) [19], which for the first time proposed multi-layer feature MAPs for detection regression. In 2017, Joseph *et al.* perfected and improved YOLO and proposed the YOLOv2 version, which greatly improved the detection speed, but was not friendly to small target detection [20]. Subsequently, Fu *et al.* Proposed Deconvolutional Single Shot Detector (DSSD) [21], introduced a residual module, and added a deconvolution layer. Compared with SSD, it has greatly improved the detection of small targets. In 2018, Joseph *et al.* proposed YOLOv3 based on the idea of SSD [22], fully optimized the algorithm, and improved the detection ability of small-scale target objects. Overall, YOLOv3 may be the most popular deep learning target detector in practical applications, because its detection accuracy and speed are well balanced.

This paper uses the YOLOv3 algorithm to study vehicle detection under road surveillance video. The analysis of the surveillance video image is shown in Figure 1. The imaging scale (number of pixels) of the vehicle target in the field of view of the surveillance camera is inversely proportional to the distance between it and the camera. The short distance is a large target and the long distance is a small target. The span of the target's imaging scale is large, which easily leads to missed detection.

In response to this problem, this article improves the YOLOv3 network. At the detection stage, considering that features such as SSD, YOLOv3, and FPN all use feature pyramid structures, this paper proposes a new multi-level feature pyramid structure added to the codec module to detect vehicle targets at different scales. First, we stitched the multi-level features extracted by the backbone network into basic features. Then, we send the above basic features to the codec module, and use the decoder layer of the codec module as the feature of the detection object. Finally, we combine the multi-level features of the backbone network with equivalent scales at the decoder layer to form a feature pyramid for target detection.

II. RELATED WORK

A. YOLOV3

YOLOv3 network evolved from YOLO and YOLOv2 networks. Compared with 2-stage detection networks such as Faster R-CNN, the YOLO network transforms the detection problem into a regression problem. In the YOLO series,

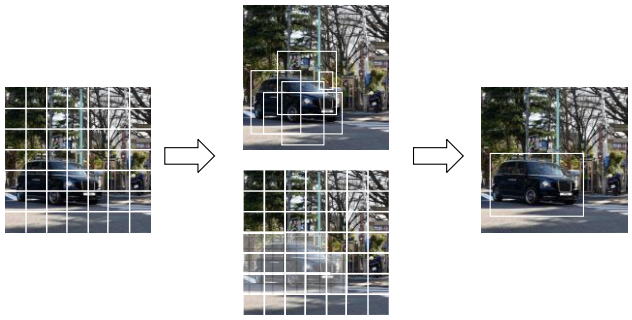


FIGURE 2. YOLOv3 flowchart.

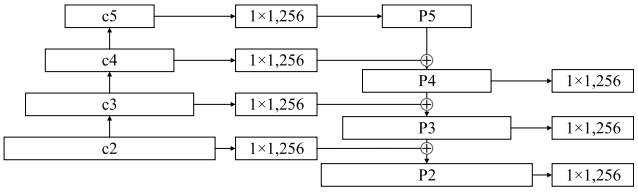


FIGURE 3. Feature pyramid.

it only needs one step to achieve the detection task. It does not need to first generate the proposed area and then detect, but directly generates the bounding box coordinates and probability of each type through regression. As such, the speed of the YOLO detection algorithm is much faster than some two-stage detectors such as the RCNN series. The YOLO detection model is shown in Figure 2. First, the network divides each image in the training set into an $S \times S$ grid. If the ground truth center of the target falls in the grid, the grid is responsible for detecting the target. Each grid is responsible for predicting the bounding boxes and their confidence scores, as well as the conditional probability. The definition is as follows:

$$Confidence = p_r\,(Object) \times IoU_{pred}^{truth},\ p_r\,(Object) \in \{0, 1\} \quad (1)$$

Among them, the confidence degree reflects whether the grid contains objects and the accuracy of predicting the bounding box when the objects are included. When the target exists in the box, $p_r\,(Object)$ is 1; when the target does not exist in the box, $p_r\,(Object)$ is 0. IoU_{pred}^{truth} indicates the degree of overlap between the prediction frame and groundtruth. When multiple bounding boxes detect the same target, YOLOv3 uses the non-maximum suppression (NMS) method [23] to select the best bounding box.

B. FEATURE PYRAMID

The image feature pyramid is a method proposed in FPN [24] to extract multi-scale information in an image. As shown in Figure 3, it is divided into three parts: a bottom-up path, a top-down path and a middle connecting part.

The bottom-up path is a feedforward calculation of the backbone network, and a feature hierarchy composed of feature MAPs of different proportions can be obtained. The top-down path is made up of higher-level feature MAPs that are more abstract in space but more semantic Sampling to generate high-level feature MAPs. Then the bottom-up

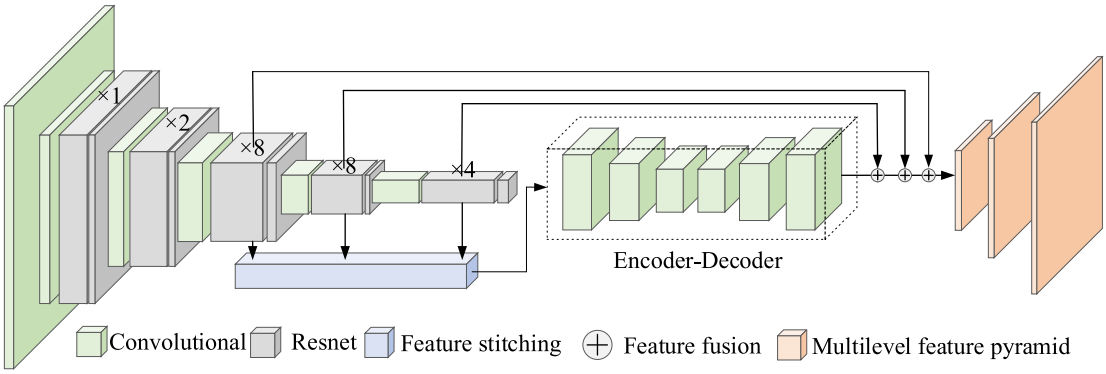


FIGURE 4. Vehicle detection model Network structure.

path is connected laterally, so that the high-level features are enhanced. Each feature MAP connecting the bottom-up path and the top-down path transversely has the same size. The low-resolution feature MAP is upsampled twice, and then the upsampling MAP and the corresponding bottom-up MAP are combined to obtain the final feature pyramid.

III. VEHICLE DETECTION ALGORITHM

Based on YOLOv3, we optimized the original network for vehicle detection under traffic surveillance video. In order to better detect multi-scale vehicles, especially small-scale vehicles, this paper proposes a new feature pyramid structure based on the codec module. First, we stitched the multi-level features extracted by the backbone network into basic features. Then, we send the above basic features to the codec module, and use the decoder layer of the codec module as the feature of the detection object. Finally, we combine the multi-level features of the backbone network with an equivalent scale at the decoder layer for detection. The improved vehicle detection model is shown in Figure 4.

The main innovations of this article are as follows:

(1) Introduce YOLOv3 algorithm to multi-scale vehicle detection under traffic video, and improve on this basis. The three-level feature MAPs generated by the backbone network are stitched by the feature stitching module to form basic features.

(2) A feature encoding and decoding structure module is proposed. This module can generate a high-order multi-scale feature MAP through a simple U-shaped structure.

(3) A special diagnosis can be integrated with a module, and an attention mechanism is added to this module, which improves the expression ability of the model.

A. BACKBONE NETWORK

This article uses YOLOv3's backbone network darknet 53. Darknet-53 has the same accuracy as Resnet-152, but its speed is more than 2 times faster than Resnet-152. The specific details are shown in Table 1.

B. FEATURE STITCHING MODULE

The feature stitching module stitches features at different levels in the backbone network as input to the encoding and decoding layer. As shown in Figure 5, the feature stitching module uses a 1×1 convolution layer to compress the channels of the input features, and uses a join operation to stitch these feature MAPs. Since the feature stitching module takes as input two feature MAPs of different scales in the backbone network, it uses an upsampling operation to rescale features at different levels to the same scale before the connection operation.

C. ENCODER-DECODER

Unlike the FPN network which selects the output of the last layer of each stage in the ResNet backbone network as its reference feature set, this paper uses a convolution operation to construct a special codec to generate a multi-level feature pyramid as shown in Figure 6. In the encoding stage, in order to generate feature reference sets of different scales, we use continuous 3×3 convolution layers to perform convolution downsampling on the input feature MAP. The decoder is a series of 3×3 convolutional layers with a step size of 1, and at the decoding stage we use the feature MAP output of each layer of the encoder as a reference feature set. In addition, we have added upsampling layers and pixel-wise summing operations on the branches of the decoder, in order to keep the feature MAPs the same size, and enhance the learning ability and maintain the smoothness of features.

D. FEATURE FUSION MODULE

The feature fusion module aims to fuse the multi-scale features generated by the codec and the multi-scale features generated by the backbone network into a multi-level multi-scale feature pyramid, as shown in Figure 3. The feature fusion module first stitches together feature MAPs with the same scale. Suppose we represent the multi-level and multi-scale feature pyramid after fusion as $P = [P1, P2, P3]$. Each scale in a multi-level, multi-scale pyramid contains depth features from each scale. Then, this paper uses feature attention mechanism [15] to aggregate features in an adaptive manner. Its specific structural details are shown in Figure 7. We take the feature fusion process at 13×13 scale as an example, and the detailed process is shown in Figure 7 above.

In the channel attention mechanism, we first embed the global information. Specifically, we first compress the global spatial information into a channel, and then perform global average pooling on the feature blocks after the flattening,

TABLE 1. Darknet-53 Structure.

Type		Filter	Channel	Stride	FeatureMAP
Convolutional		3×3	32	1	416×416
Convolutional		3×3	64	2	208×208
Convolutional		1×1	32	1	
Convolutional	1	3×3	64	1	
Residual					208×208
Convolutional		3×3	128	2	104×104
Convolutional		1×1	64	1	
Convolutional	4	3×3	128	1	
Residual					104×104
Convolutional		3×3	256	2	52×52
Convolutional		1×1	128	1	
Convolutional	8	3×3	256	1	
Residual					52×52
Convolutional		3×3	512	2	26×26
Convolutional		1×1	256	1	
Convolutional	8	3×3	512	1	
Residual					26×26
Convolutional		3×3	1024	2	13×13
Convolutional		1×1	512	1	
Convolutional	4	3×3	1024	1	
Residual					13×13

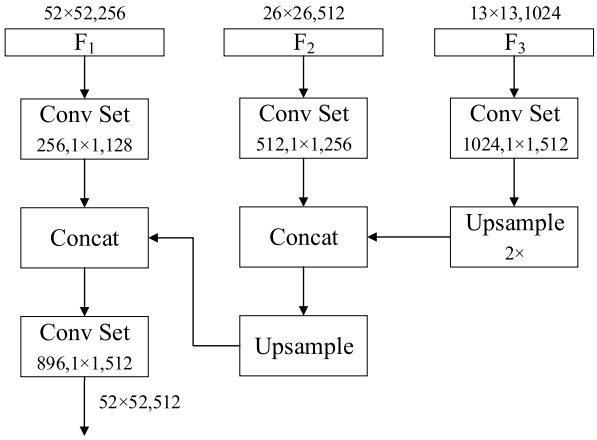


FIGURE 5. Feature stitching module.

which is the information compression operation in the figure. Formally, statistics $x \in R^C$ are generated by shrinking P by $W \times H$ in the spatial dimension, where the c-th element of x is calculated by the following formula:

$$x_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} p_c(i, j) \tag{2}$$

In order to make use of the information gathered in the compression operation, we next fully capture the channel dependencies through information activation operations, both

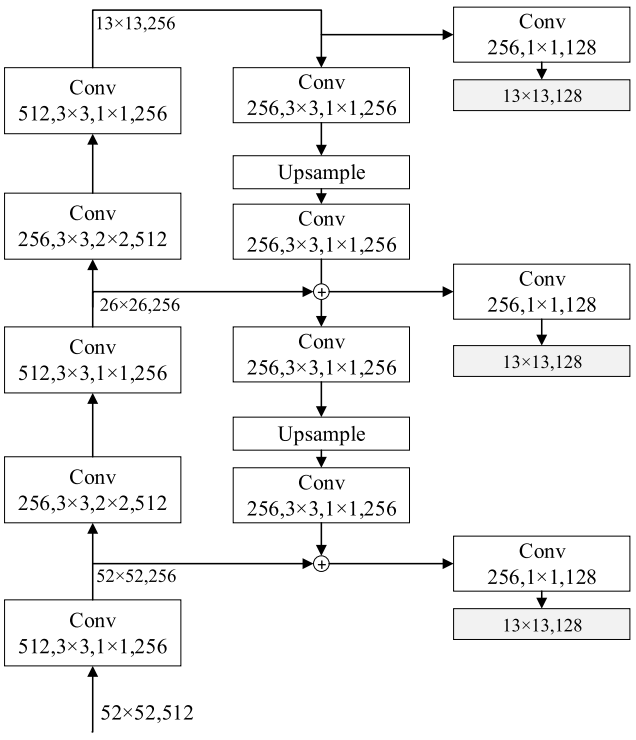


FIGURE 6. Encoder-decoder.

aggregate features:

$$s = \mathrm{F}_{ex}(x, W) = \sigma(W_2 \delta(W_1 x)) \tag{3}$$

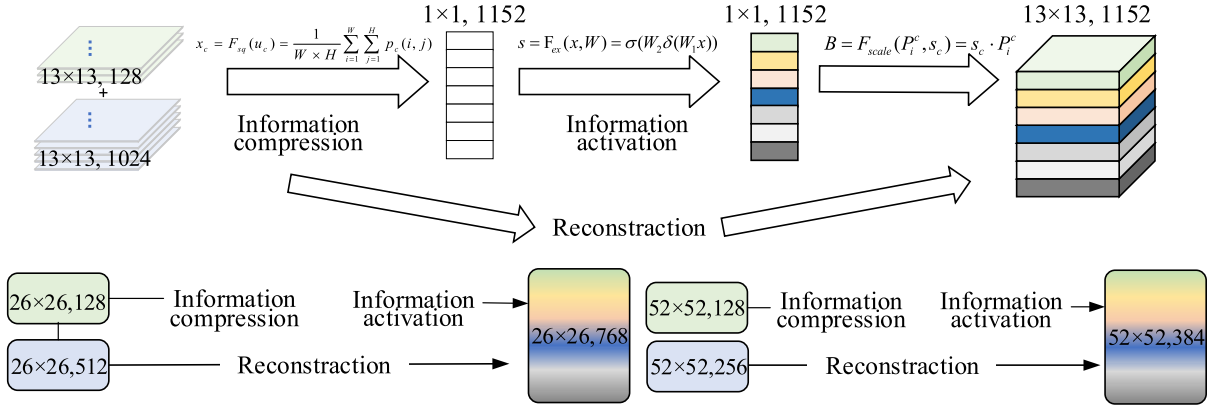


FIGURE 7. Feature fusion module.

Among them, σ represents the ReLU function: $\sigma(x) = \begin{cases} \lambda x_i, & x > 0 \\ 0, & x < 0 \end{cases}$, δ represents the sigmoid function: $\delta(x) = \frac{1}{(1+e^{-x})}$; $W_1 \in R^{\frac{C}{r} \times C}$, $W_2 \in R^{C \times \frac{C}{r}}$, and r is the reduction ratio. We use two fully connected layers as the gate mechanism, that is, the dimension reduction layer parameter is $W1$, the channel Attention Module dimension reduction ratio is r(r is 16 in the experiment), and the ReLU is followed by an ascending dimension layer with parameters $W2$. Finally, we re-weight output B for x:

$$B = F_{scale}(P_i^c, s_c) = s_c \cdot P_i^c \tag{4}$$

E. LOSS FUNCTION DESIGN

We have not made many modifications to the loss function of YOLOv3. The loss function of the algorithm in this paper is divided into three parts, one is the bounding box coordinate error, the bounding box confidence error, and the classification error. Its loss function is shown in equation (4):

$$\begin{aligned} Loss = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (2 - w_i \times h_i) [(w_i - \hat{w}_i)^2 \\ & + (h_i - \hat{h}_i)^2] - \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [\hat{C}_i \log(C_i) \\ & + (1 - \hat{C}_i) \log(1 - C_i)] \\ & - \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [\hat{C}_i \log(C_i) \\ & + (1 - \hat{C}_i) \log(1 - C_i)] - \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes}^{S^2} \\ & \times [\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))] \end{aligned}$$

In the formula, the first term is the loss function of the center coordinate of the bounding box, the second term is the loss function of the height and width of the bounding box, the third term is the loss function of the confidence of the bounding box with the object, and the fourth term is the non-existing object. The bounding box confidence loss function, the fifth term is the classification error of the element grid of the object. S is the cell grid division coefficient of the picture, B is the number of bounding boxes predicted by each grid, C is the total number of classifications, and p is the class probability. It means that there is an object in the i-th cell grid, and the j-th bounding box in the cell predicts the target. And are weight coefficients for different loss functions.

IV. EXPERIMENT AND ANALYSIS

The vehicle detection model in this paper is based on publicly available Darknet53 and Pytorch of YOLOv3. In the experiments in this article, a deep learning server with a CPU of i7-9700 GPU and NVIDIA GTX 1080ti was used. In this section, our backbone network is first pre-trained on the ImageNet 2012 dataset [25], then the entire network is trained on the KITTI [26] training set, and tested on the KITTI test set. In the experiment, the dimension reduction ratio in the channel attention module is set to 16. As for the input size, it follows the original YOLOv3 network, that is, 416×416. During detection, this paper also uses clustering to generate 9 prior frames, and finally uses non-maximum value suppresses post-processing, leaving a more accurate vehicle frame.

A. DATASET

1) KITTI DATASET

The KITTI data set was taken while driving in rural areas of Karlsruhe in the medium-sized city and around highways. The characteristics of a single image in the data set are very similar to those of video images under surveillance video. Its data set is shown in Figure 8.

2) UA-DETRAC DATASET

UA-DETRAC is a challenging real-world multi-object detection and multi-object tracking benchmark. The dataset consists of 10 hours of videos captured with a Cannon EOS 550D camera at 24 different locations at Beijing and Tianjin in China. The videos are recorded at 25 frames per seconds (fps), with resolution of 960 × 540 pixels.

FIGURE 8. KITTI dataset.

FIGURE 9. UA-DETRAC dataset.

TABLE 2. Average precision in three different difficulty levels under the KITTI dataset.

Algorithm name	Average Precision (AP)/ %			Time
	Easy	Moderate	Hard	
DPM	87.23	77.46	61.12	110
R-CNN	32.23	26.04	20.93	-
Faster R-CNN	87.90	79.11	70.19	142
MMLab-PointRCNN	95.22	91.90	87.11	100
RefineNet	90.16	79.21	65.71	200
LTN	94.68	91.18	81.51	
MonoGRNet	88.65	77.94	63.31	40
Pointpillars	88.35	86.10	79.83	16
CFENet	93.91	93.26	86.99	-
Aston-EAS	93.91	91.02	77.93	240
ARPNET	94.00	90.99	83.49	80
YOLOv3	92.55	88.71	77.78	28
Our Method	95.04	92.39	87.51	34

B. ANALYSIS

1) RESULT ON KITTI BENCHMARK

In order to verify the effectiveness of the algorithm in this paper, the algorithm in this paper and DPM, R-CNN, Faster R-CNN, MMLab-PointRCNN [27], RefineNet [28], LTN [29], MonoGRNet [30], PointPillars [31], Aston-EAS [23], ARPNET, and YOLOv3 performed comparative tests in their KITTI test set. Figure 9 is the P-R graph of the above algorithms in three different index tests of the KITTI dataset. Table 2 is the average precision rate of three different indicators under the KITTI data set. Figure 10 shows some of the detection results in the KITTI dataset.

KITTI data set is a real image taken by vehicle camera on the road, which has a high similarity with the image under the traffic video. As can be seen from Figure 9 and table 2, the AP [32] of the algorithm in this paper under three different standards of KITTI data set reaches 95.04%, 92.39% and 87.51% respectively, which are improved compared with YOLOv3, respectively: 2.49%; 3.68%; 9.73%. Because in the YOLOv3 detection model, it just up-samples the convolutional feature MAP at the bottom up to 2 times,

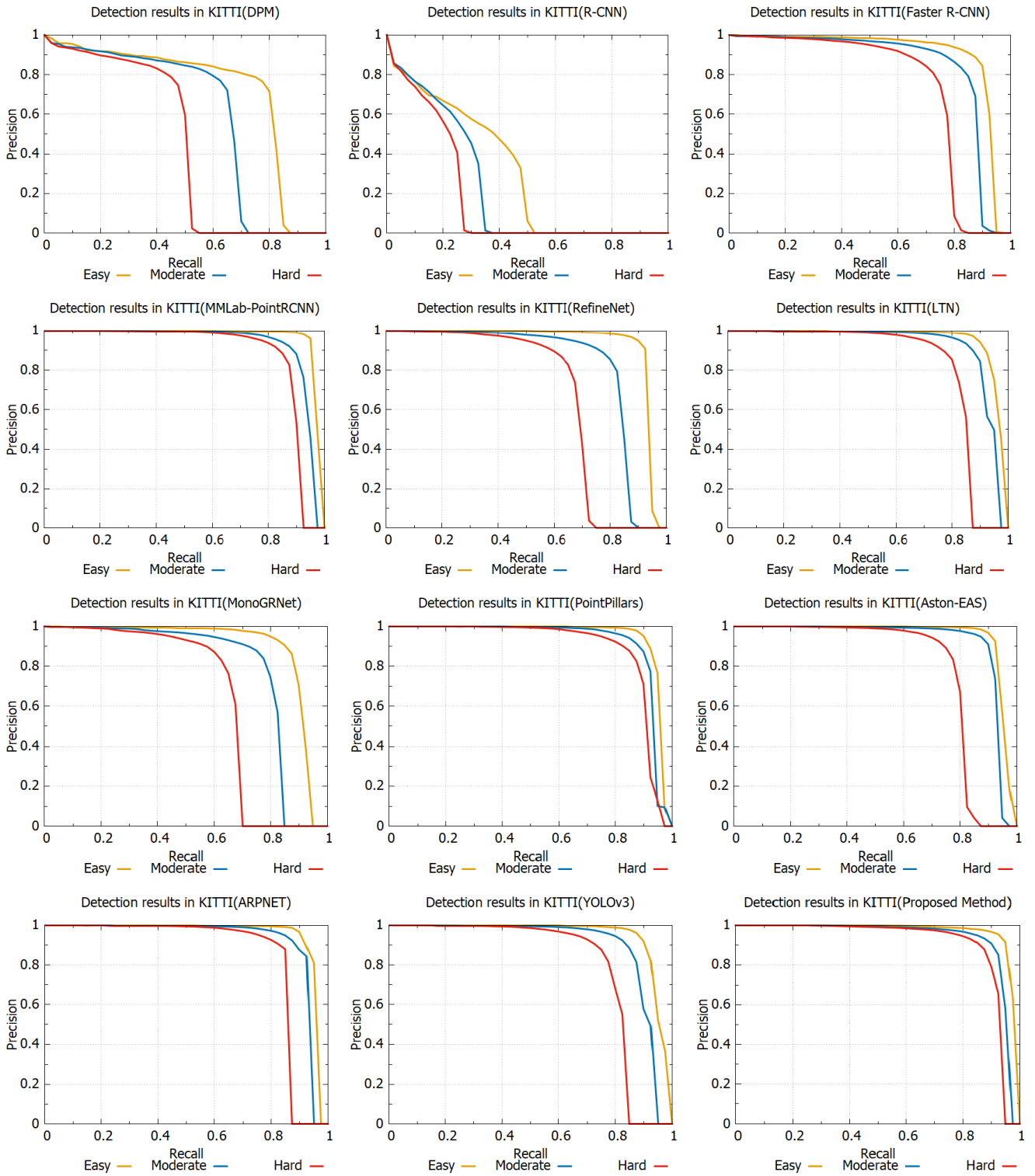


FIGURE 10. P-R diagrams in three different difficulties under the KITTI dataset.

and then performs feature stitching with its upper stage. In our vehicle detection model, we first generate the data from the backbone network. The three-scale feature MAPs are fused to form a basic feature MAP that is passed into a U-shaped codec to generate a higher-order multi-scale feature MAP. Then, we use the multi-scale feature MAP generated in the U-codec and the three-scale feature MAP generated in the original backbone network to perform feature fusion and add a feature attention module to improve the feature expression ability. Compared with YOLOv3, this model has higher feature expression ability, can better find small targets in the detection picture, and can generate more

FIGURE 11. The test results of this algorithm are used on the KITTI test set. The red circles circle the vehicle detection results at a small scale. It can be seen that the algorithm in this paper can better adapt to the occlusion of the target and has a strong adaptability to the target's scale Capability, the detection of small targets is relatively stable, and it can better meet the actual needs of vehicle detection under complex traffic video.

semantic information, thereby further improving the overall detection effect. It can also be seen that compared to two-stage target detection algorithms such as R-CNN and Faster R-CNN, the algorithm in this paper is based on the improvement of the first stage target detection algorithm of yorov3, and treats the target detection process as a regression problem. Unlike R-CNN series, which first generates a large number of candidate frames using the regional recommendation network (RPN), and then carries out target recognition, the speed will be greatly improved, and the accuracy is also higher than R-CNN series The improvement is mainly due to the combination of shallow spatial information and deep semantic information.. Compared with the current popular vehicle detection algorithm, the algorithm in this paper compared with such as LTN algorithm, MonoGRNet algorithm, Aston EAS algorithm, ARPNET algorithm has a certain degree of accuracy improvement, and the speed is also in the priority. Compared with MMLab-PointRCNN, although the accuracy of the algorithm in this paper is not as good as the former under Easy difficulty, the speed is almost three times that. Compared with the Pointpillars algorithm, although the Pointpillars algorithm is superior in speed, it is possible that our method No error will occur in practical applications. We believe that the algorithm in this paper is more effective when dealing with traffic video vehicle detection.

2) RESULT ON UA-DETRAC BENCHMARK

In order to verify the robustness of the algorithm in this paper, we decided to choose the same experimental method

TABLE 3. Average precision under 8 different conditions in UA-DETRAC data set.

Method	Average Precision(AP)/ %							
	full	easy	medium	hard	cloudy	night	rainy	sunny
DPM	25.70	34.42	30.29	17.62	24.78	30.91	25.55	31.77
R-CNN	48.95	59.31	54.06	39.47	59.73	39.32	39.06	67.52
Faster R-CNN	57.72	80.33	62.25	42.44	57.97	62.20	47.84	69.75
SA-FRCNN	45.83	73.93	49.00	30.76	49.97	52.30	33.39	55.04
YOLOv3	63.45	84.94	68.05	49.25	67.34	67.50	50.16	74.85
Our Method	68.01	88.28	73.04	55.73	72.00	69.53	70.89	78.89

TABLE 4. Average precision in three different difficulty levels under the KITTI dataset.

Structure name			Average Precision (AP)/ %		
Feature stitching	Encoding and decoding	Feature fusion	Easy	Moderate	Hard
	√	√	92.94	90.83	83.81
		√	90.86	88.02	80.29
√	√		93.04	92.13	84.81
√	√	√	94.16	93.52	86.71

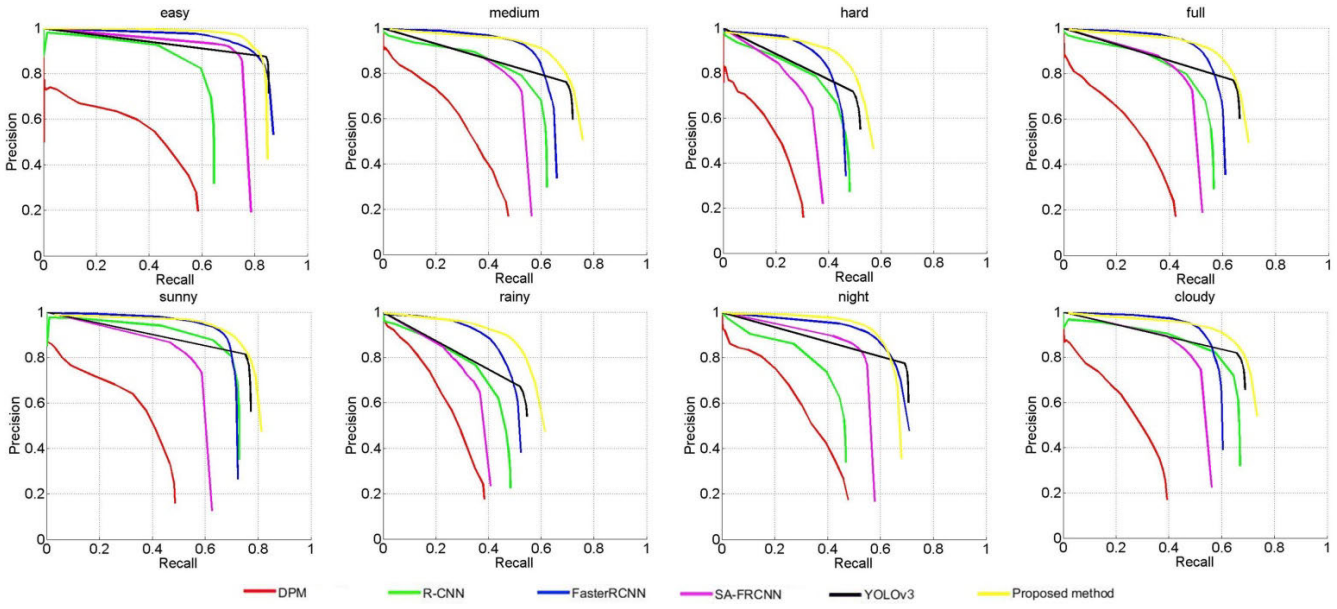


FIGURE 12. P-R diagrams in different difficulties under the UA-DETRAC dataset.

as that under the KITTI data set, but the selected data set is UA-DETRAC: the comparison methods we chose are all the detection methods recorded on UA-DETRAC. Different from the KITTI test benchmark, there are eight conditions on UA-DETRAC, namely, easy, medium, hard, full, sunny, rainy, night and cloudy. Figure 11 and table 3 show the performance under eight different conditions in the UA-DETRAC data set. It can be seen from table 3 that the detection effect of traditional vehicle detection algorithm DPM in complex environment is poor, even in easy condition, the average precision rate is not up to 35%. Starting from R-CNN, the average precision rate of vehicle detection based on deep learning algorithm in complex environment, such as in traffic video, is better than that of traditional algorithm. Under sunny, the average precision rate can to 69.75%. Compared with YOLOv3, the algorithm in this paper improves obviously in every detection condition.

C. MODEL ANALYSIS

The algorithm in this paper is composed of different modules, so we need to verify the effectiveness of each individual module on the final performance. Because the data volume of KITTI dataset is too large, and it costs a lot to carry out multi group splitting experiments, so in the next experiment, we randomly selected 500 datasets from KITTI dataset as model analysis experiments, according to the ratio

of 4:1 Examples are divided into training set and test set. We carry out comparative experiments on feature splicing module, encoding and decoding module and feature fusion module respectively, and the results are shown in Table 4.

1) FEATURE STITCHING MODULE

In order to verify that the feature stitching has a positive impact on the final results of the model, this article will not use the feature stitching module and directly input the final layer of network feature MAPs into the encoding and decoding module. The results are different from the use of the feature stitching module for the backbone network Depth feature MAPs are fused and passed to the codec module for comparison. Through rows 1 and 4 of the table, we can clearly see that the average precision rates under three different difficulties have increased by 1.22%, 2.69%, and 2.90%, respectively.

2) ENCODING AND DECODING MODULE

The function of the encoding and decoding module is to encode and decode the feature MAP after fusion to generate multi-scale features. In order to verify the effectiveness of the encoding and decoding module, we did a set of comparative experiments. We connected three sets of feature MAPs directly generated by the backbone network to the feature fusion module instead of using the multi-scale feature MAPs generated by it through the table. It can be clearly seen that the average precision is reduced in the second and fourth lines.

3) FEATURE FUSION MODULE

From the 3rd and 4th rows in the table, when the model is added with the feature fusion module, the average precision rate under all difficulties becomes more accurate.

V. CONCLUSIONS

In this paper, the YOLOv3 network model is introduced into the field of vehicle detection under traffic monitoring videos. It is found that in the actual detection process, small-scale vehicles often miss detection. On the basis of YOLOv3, in order to efficiently and accurately generate multi-scale features to adapt to the detection of multi-scale target vehicles, we propose a new feature pyramid module based on encoding and decoding. First, we stitched the multi-level features extracted by the backbone network into basic features. Then, we send the above basic features to the codec module, and use the decoder layer of the codec module as the feature of the detection object. Finally, we combine the multi-level features of the backbone network with equivalent scales at the decoder layer to form a feature pyramid for target detection. Tested on the KITTI dataset, the effect has been improved. Good detection results have been achieved for vehicle targets of various scales, especially for small target detection. The accuracy is significantly improved than the YOLOv3 algorithm, which can better meet the needs of practical applications.

REFERENCES

[1] P. Felzenszwalb, D. McAllester, and D. Ramaman, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE CVPR*, Jun. 2008, pp. 1–8.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[4] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[5] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[6] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.

[7] L.-Q. Zuo, H.-M. Sun, Q.-C. Mao, R. Qi, and R.-S. Jia, "Natural scene text recognition based on encoder-decoder framework," *IEEE Access*, vol. 7, pp. 62616–62623, 2019.

[8] Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Fast and efficient non-contact ball detector for picking robots," *IEEE Access*, vol. 7, pp. 175487–175498, 2019.

[9] W. Q. Zheng, J. S. Yu, and Y. X. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *Proc. Int. Conf. Affect. Comput. Intell. Interact. (ACII)*, Sep. 2015, pp. 827–831.

[10] H. Zhang, Y. Fu, L.-B. Feng, Y. Zhang, and R. Hua, "Implementation of hybrid alignment algorithm for protein database search on the SW26010 many-core processor," *IEEE Access*, vol. 7, pp. 128054–128063, 2019.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

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

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[20] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 7263–7271.

[21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–6.

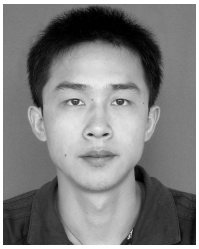
[22] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 390–399.

[23] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2006, pp. 850–855.

[24] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, Jul. 2017, pp. 936–944.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[27] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
[28] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.
[29] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1355–1361.
[30] Z. Qin, J. Wang, and Y. Lu, "MonoGRNet: A geometric reasoning network for monocular 3D object localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8851–8858.
[31] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.
[32] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2006, pp. 102–111.



FENG HONG was born in Chizhou, Anhui, China, in 1983. He received the M.S. degree in testing technology and automation equipment from the Guilin University of Technology, in 2011, and the Ph.D. degree in signal and information processing from the Hefei University of Technology, in 2017. Since 2011, he has been with Chizhou University. His research interests include target detection and intelligent information processing.

CHANG-HUA LU was born in Chaohu, Anhui, in 1962. He received the bachelor's degree in engineering in 1983, the master's degree in 1988, and the Ph.D. degree from the Graduate School, Chinese Academy of Sciences, in 2001. He is currently a Professor and a Ph.D. Tutor with the Hefei University of Technology. He is also a Ph.D. Tutor with the Hefei Material Branch of the Chinese Academy of Sciences. His research interests include signal detection and processing, computer applications, photoelectric information processing, automatic test systems, and multimedia information transmission technologies.

CHUN LIU was born in 1966. She received the Bachelor of Engineering degree in automation in April 1996 and the master's degree in engineering from Zhejiang University. She has been an Associate Professor, a master's tutor for the Bachelor of Engineering with the Harbin University of Science and Technology, since July 1988. Her research interests include DSP technology applications, and testability and fault diagnosis technologies.

RU-RU LIU was born in 1986. She received the degree major in traffic information engineering and control from Shanghai Maritime University, in 2011. Since 2012, she has been working as an Associate Professor with Chizhou University. Her research interests include intelligent information processing and semiconductor performance testing.

JU WEI received the B.S. degree from Southern Medical University, in 2009, the M.S. degree from the Chongqing University of Technology, in 2012, and the Ph.D. degree in signal and information processing from the School of Computer and Information Engineering, Hefei University of Technology. She is currently a Lecturer with the School of Internet, Anhui University. Her research interests include intelligent information processing, optical communication, and pattern recognition.