IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 October 2023, accepted 21 October 2023, date of publication 26 October 2023, date of current version 3 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327735

RESEARCH ARTICLE

Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO

ZHENYANG LV[1], RUGANG WANG[1], YUANYUAN WANG[1], FENG ZHOU[1], AND NAIHONG GUO[2]

[1]School of Information Technology, Yancheng Institute of Technology, Yancheng 224051, China
[2]Yancheng XiongYing Precision Machinery Company Ltd., Yancheng 224006, China

Corresponding author: Rugang Wang (wrg3506@ycit.edu.cn)

This work was supported in part by the Jiangsu Graduate Practical Innovation Project under Grant SJCX22_1685, in part by the Major Project of Natural Science Research of Jiangsu Province Colleges and Universities under Grant 19KJA110002, in part by the Natural Science Foundation of China under Grant 61673108, and in part by the Natural Science Research Project of Jiangsu University under Grant 18KJD510010.

ABSTRACT To address issues such as low detection accuracy and limited real-time performance in road scene detection, a novel road scene detection algorithm based on CMS-YOLO is proposed in this paper. In this algorithm, an efficient backbone called the cross-stage partial DWNeck is devised. By using large-scale depthwise separable convolutions and residual structures, it enables the acquisition of more comprehensive feature information, thereby increasing both the receptive field and the richness of extracted features. Meanwhile, a feature pyramid called the multi-scale fusion feature pyramid network is designed to strengthen the fusion of shallow and deep-level information, effectively preventing the loss of feature information in the transmission process. Besides, a new decoupled head structure called the special decoupled head is introduced, which effectively addresses the conflict between classification and regression tasks through a three-layer joint output structure. Finally, experiments were conducted on two publicly available datasets, namely Udacity Self-Driving and BDD100K. Experimental results indicate that the CMS-YOLO algorithm achieved an impressive detection accuracy of 90.3% and 59.1% in mAP@0.5, demonstrating a remarkable improvement of 4.2% and 7.2% over YOLOv5 respectively. Moreover, in real-world scenarios, the algorithm achieves an impressive real-time detection speed of 34.5 frames per second. These results demonstrate that CMS-YOLO not only meets but also surpasses the requirements for detection accuracy and real-time performance for object detection in autonomous driving scenarios.

INDEX TERMS Autonomous driving, road scene detection, YOLO, cross stage partial DWNeck, multiscale fusion feature pyramid networks, special decoupled head.

I. INTRODUCTION

In recent years, the rapid development of deep learning has led to the widespread adoption of object detection techniques across various domains, attracting considerable interest in the field of autonomous driving vehicles. The integration of cameras with object detection algorithms provides a more accurate and cost-effective approach to object recognition, making it highly suitable for large-scale deployment in future autonomous driving systems [1], [2], [3]. Therefore, the current research focuses on developing object detection algorithms that strike a balance between real-time performance and high detection accuracy, meeting the specific requirements of autonomous driving applications. In this context, deep learning-based object detection algorithms have emerged as a crucial and prominent research area [4].

Within the scope of the current research, target detection algorithms based on convolutional neural networks can be mainly categorized into two types: one-stage and two-stage detection algorithms. The first category, represented by YOLO and SSD, employs a regression strategy to facilitate target detection by bypassing the region proposal phase and directly regressing the classification and bounding boxes of the targets [5], [6]. The second category, represented

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

by R-CNN, SSP-Net, and Fast R-CNN, initially traverses the entire image to obtain the proposed region boxes and subsequently performs classification and detection tasks on the targets. Though these two-stage detection algorithms exhibit higher accuracy in target detection, they tend to have slower detection speeds. Consequently, their applicability in autonomous driving applications is limited [7], [8], [9]. One-stage detection algorithms are end-to-end algorithms that utilize bounding box regression. These algorithms have advantages in real-time performance, albeit at the cost of a slight reduction in detection accuracy. However, they face challenges in detecting small objects and tend to cause missed detections [5]. In recent years, J. Redmon et al. have made remarkable progress in improving the YOLO algorithm. They have significantly enhanced the algorithm's detection accuracy by integrating the anchor mechanism, incorporating the feature pyramid network (FPN), and replacing the backbone. However, the improved algorithm still cannot meet the specific application requirements of industrial scenarios, indicating that further improvements are needed [10], [11]. In 2020, A. Bochkovskiy et al. made notable progress in the YOLO algorithm. They introduced the FPN+PAN structure as the Neck component and replaced the Backbone with the CSPDarknet-53 architecture. These changes substantially enhanced both detection speed and accuracy [12]. Drawing on this progress, in 2021, Ge et al. addressed the issue of classification and regression conflict by replacing the coupled head with a decoupled head, leading to a significant improvement in detection accuracy [13]. In 2022, Li et al. made a great contribution by incorporating the RepVGG structure into YOLO, thus substantially improving detection speed [14]. In the same year, Wang et al. elevated the YOLO series algorithms for industrial applications. They integrated the E-ELAN as the Backbone and fused the MP structure for downsampling, which greatly enhanced detection accuracy [15].

With the continuous advancement of deep learning-based object detection algorithms, more and more exceptional algorithms are being applied to autonomous driving. For example, Chen et al. proposed the DW-YOLO algorithm, which aims to improve the detection accuracy of road scene objects by increasing the depth and width of the network. However, this approach involves a large number of parameters, which can hinder real-time performance [16]. H. Wang et al. proposed the MobileNet-YOLOv4 algorithm, which has been applied to autonomous driving to enhance detection speed. However, it incurs a loss in detection accuracy and still has difficulties in detecting small objects, leading to missed detections or false alarms [17]. Cai et al. introduced YOLOv4-5D and applied it to autonomous driving. This approach utilizes five scale detection layers and replaces the original backbone network with DCN, leading to improved detection accuracy for small objects. However, it suffers from a large parameter count and reduced real-time performance [18]. Consequently, finding a balance between detection accuracy and speed has become a significant research focus in the field.

To address the above issues in existing algorithms, this paper proposes the CMS-YOLO algorithm. Firstly, it introduces the Cross-stage partial DWNeck (CD) module to replace the original C3 module in the network backbone. The CD module utilizes larger convolutional kernels and more residual connections, leading to a larger receptive field and richer feature information compared to the C3 module. Secondly, a Multi-scale Fusion Feature Pyramid Network (MFFPN) is proposed to replace the FPN+PAN structure in the original network. The MFFPN effectively integrates contextual and feature information across different feature layers, which helps to enhance the richness of feature information and mitigate information loss during propagation. Additionally, a novel detection head structure called Special Decoupled Head (SDH) is developed to replace the coupled head structure in the original network. The SDH adopts a three-layer joint output structure for decoupling, thereby effectively addressing the conflict between localization and classification tasks. Compared to the decoupled head structure in YOLOX, the SDH demonstrates better decoupling capability and detection accuracy. Overall, the proposed CMS-YOLO algorithm achieves higher detection accuracy while maintaining a fast detection speed.

II. MODEL ARCHITECTURE

The YOLOv5 model has many variants in terms of depth and width, including YOLOv5-s, YOLOv5-m, YOLOv5-l, and YOLOv5-x. Considering the real-time requirements, YOLOv5-s is selected in this paper as the base model for modifications. The overall structure of YOLOv5-s consists of four main components: Input, Backbone, Neck, and Head. The Input component incorporates various data augmentation techniques, such as adaptive image scaling and translation, random horizontal flipping, and mosaic. Meanwhile, it involves the calculation of adaptive anchor boxes using K-means. The Backbone comprises the CBS module, C3 module, and SPPF spatial pyramid pooling module. These modules utilize a series of convolutional operations to extract feature representations from the input images. The Neck network adopts the FPN and PAN (Path Aggregation Network) structures to fuse the extracted image features through path aggregation, thereby achieving effective feature fusion. The Head component comprises three detection heads, each corresponding to an object scale: large, medium, and small. The output of the detection heads includes class probabilities, classification information, and coordinate information of the detected objects. Overall, the YOLOv5-s model combines these components to realize efficient and accurate object detection, making it suitable for various applications.

In this section, the CMS-YOLO architecture is proposed, which builds upon the traditional YOLOv5 network structure and optimizes the C3 module, feature fusion structure, and

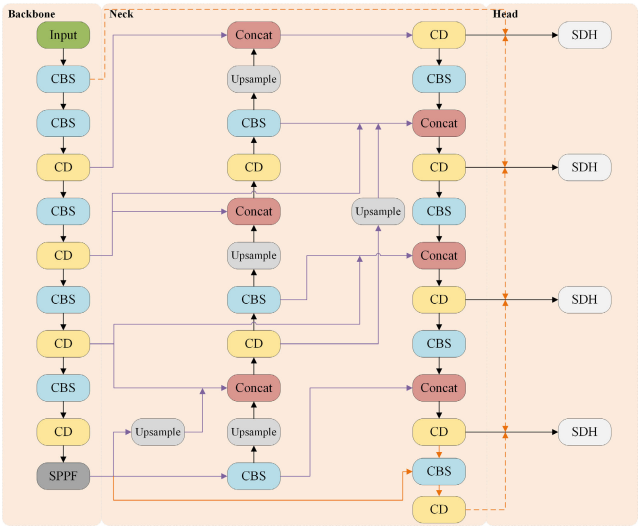

FIGURE 1. The Architecture of the CMS-YOLO Network. In the diagram, the CBS module consists of a 3×3 ordinary convolution layer, a batch normalization (BN) layer, and the SiLU activation function. The CD module represents the backbone network module proposed in this study. The light purple lines and connected upsampling depict the feature fusion structure known as MFFPN, introduced in this paper. The SDH module represents the proposed detection head, and the orange lines, along with the final CBS and CD modules in the Neck, indicate the output of the SDH. It is noteworthy that these components do not participate in feature extraction and fusion within the network.

detection heads. The objective is to overcome the challenges encountered in detecting road scenes for autonomous driving, such as low detection accuracy and poor real-time performance. The CMS-YOLO architecture is illustrated in Fig.1 As shown in this figure, the C3 module is replaced with the CD module (depicted as the golden block). This replacement leads to a larger receptive field and facilitates the extraction of more semantic features, thereby preparing the network for subsequent feature fusion. Meanwhile, the FPN+PAN structure is replaced with the MFFPN structure (represented by the light purple lines and connected upsampling), enabling a comprehensive fusion of shallow and deep-level feature information. This fusion improves the recognition accuracy of small objects in road scenes. Besides, the three detection heads are replaced with four detection heads, and the original coupled head is replaced with the SDH detection head (depicted as the white block in the figure). The orange lines in the figure represent the output of the SDH, which are separate from the feature extraction and fusion process within the network. This replacement allows the effective use of feature information from both higher and lower layers to address the conflict between classification and localization tasks, thereby improving detection accuracy for challenging objects in road scenes.

To sum up, by incorporating optimizations in the C3 module, feature fusion structure, and detection heads, the CMS-YOLO architecture improves detection accuracy and real-time performance for road scene detection in intelligent driving applications.

A. THE CD MODULE

In road scene detection tasks, dense objects usually occupy a limited number of pixels and are susceptible to background interference. The C3 module in YOLOv5 is prone to losing important feature information or being affected by background pixels during feature extraction. The proposed CD module in this paper can address these issues effectively. The CD module has a larger receptive field, more efficient residual connections, and superior feature extraction capabilities, all while maintaining a minimal number of parameters. When applied to road scene object detection, the CD module mitigates issues related to the loss of feature information and interference from background pixels, significantly enhancing the accuracy of detecting densely packed or obscured vehicles and pedestrians. As shown in Fig. 2, all BottleNeck components in the C3 module are replaced with the proposed CD module, which mitigates the loss of critical features and minimizes the impact of background pixels.

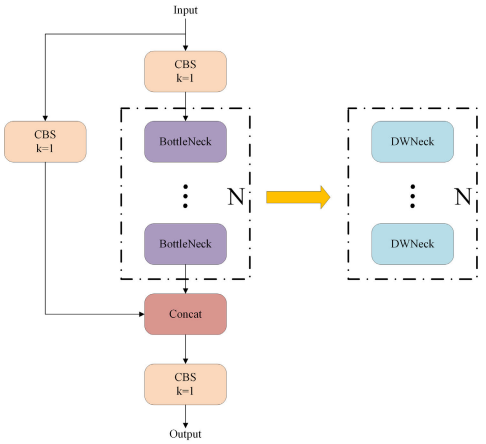

FIGURE 2. The Structure of the Proposed CD Module. The CD module structure showcases the replacement of the BottleNeck modules in the C3 module with the proposed DWNeck module.

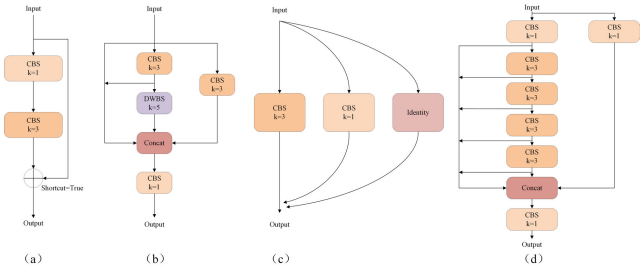

FIGURE 3. Different Base Modules. (a) BottleNeck: The base module used in YOLOv5. (b) DWNeck: The base module in the network proposed in this paper. (c) RepConv: The base module used in YOLOv6. (d) E-ELEN: The base module used in YOLOv7. The figure illustrates the distinctive structural characteristics of the base modules used in different versions of the YOLO series.

Having a larger receptive field in the network helps to capture more comprehensive contextual feature information,

thereby enhancing the detection accuracy of dense objects in road scene detection tasks. However, large kernel convolutions are computationally intensive, which limits their practical use in real-time detection tasks [19], [20]. Recent studies have shown that it is possible to expand the receptive field by using large-kernel depthwise separable convolutions (DW) while minimizing computational costs [21]. Inspired by these findings, this paper introduces the CD module as a replacement for the BottleNeck modules in the C3 module of the original network. As illustrated in Fig.3 (b), the CD module incorporates a 5×5 DW convolution to increase the receptive field of the base module. The 1×1 convolutions are replaced with 3 × 3 convolutions, and an additional 3×3 convolution is introduced with an extra residual connection on the residual branch. The output of the first 3×3 convolution serves as a residual branch output, and the Add operation is replaced with the Concat operation. Additionally, a 1×1 convolution is added to adjust the output channel count. This approach enables a more comprehensive extraction of contextual information, leading to a significant increase in detection accuracy.

Compared to the module in Fig. 3(a), the module proposed in this paper (Fig. 3(b)) provides a significantly larger receptive field, enabling the extraction of more comprehensive feature information and establishing a solid foundation for subsequent feature fusion. Compared to the module in Fig. 3(d), the proposed module uses a similar residual concept but reduces the number of 3×3 convolutions. The use of two 3×3 convolutions and one 5×5 DW convolution for feature extraction effectively reduces the parameter count while preserving the richness of extracted features. Meanwhile, the module in Fig. 3(c) represents the currently popular reparameterization structure, which is considered a tool for improving detection accuracy with no additional cost. However, it introduces challenges such as longer training time, higher hardware consumption, and larger quantization errors after model compression. In comparison, the module proposed in this paper provides a simpler and more efficient alternative. It requires fewer training resources and exhibits smaller quantization errors after model compression.

B. MFFPN

In YOLOv5, the FPN+PAN feature pyramid structure is utilized to fuse shallow-level and deep-level feature maps, thereby combining their rich semantic and positional information for road scene detection. However, this approach suffers from information loss in the unidirectional feature enhancement process. To address this issue, this paper proposes the MFFPN. Illustrated in Fig. 4 and represented by the light purple lines in Fig. 1, the MFFPN structure provides several enhancements. The MFFPN structure addresses the problem of extracting feature information from small objects within the FPN+PAN structure. By employing an up-sampling feature fusion approach, it resolves the issue of feature loss commonly encountered in traditional

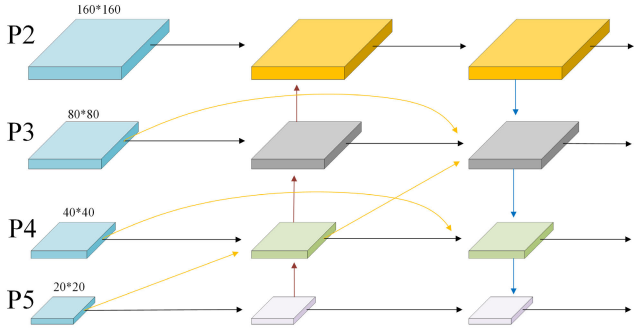

FIGURE 4. The Structure of the Multi-scale Fusion Feature Pyramid Network. The red lines form the FPN structure, transmitting deep feature information to shallower layers, enhancing semantic representation at multiple scales. The blue lines constitute the PAN structure, transmitting shallow information to deeper layers, enhancing localization capabilities. The golden lines fuse shallow and deep information, addressing information loss issues and augmenting the diversity of feature information.

convolutional down-sampling feature fusion. In the task of road scene object detection, the MFFPN structure greatly improves the accuracy of detecting small objects while ensuring the integrity of information during feature fusion.

The MFFPN structure introduces an additional output of size 160×160, thereby expanding the original three-level feature map output to four levels. This expansion enables capturing more detailed feature information related to small and densely packed objects in the shallow-level feature maps. Meanwhile, the shallow-level feature information of sizes 80×80 and 40×40 is directly connected to the output and fused with the deep-level feature information. Besides, the shallow-level feature information of size 20×20 is fused with the deep-level feature information of size 40×40 and 80×80, as indicated by the golden section in the figure. This comprehensive fusion of shallow-level and deep-level features effectively addresses the issue of information loss, leading to improved detection accuracy. It is important to note that the MFFPN structure diverges from the fully connected structure between shallow-level and deep-level feature maps used in BiFPN. This deviation is mainly due to concerns about parameter overload caused by multiple Concat operations. Such concerns make the fully connected structure less suitable for autonomous driving applications. In contrast, the proposed MFFPN structure achieves a balance between feature fusion and computational efficiency, which improves the detection accuracy for small and densely packed objects in road scenes.

C. SPECIAL DECOUPLED HEAD

In recent years, the decoupled head structure has become a preferred choice in object detection algorithms, such as YOLOX, YOLOv6, YOLOv8, and DOOD [13], [14], [22]. YOLOX introduces the decoupled head structure to the YOLO series to solve the problem of conflicting attention between classification and regression tasks in the coupled head structure. This has significantly improved both the

detection accuracy and convergence speed of the algorithms. In addition to YOLOX, YOLOv8 further optimizes the decoupled head structure by introducing a new approach that eliminates the Objectness branch, thereby achieving higher detection accuracy. In this approach, deformable convolutions are utilized with learnable offsets in each branch to enable the adaptive selection of spatial features for each output detection head [22]. However, it is noteworthy that most existing algorithms still rely on the conventional decoupled head structure. Although these algorithms utilize separate parameters to learn task-specific features, they fail to fundamentally address the inherent conflict between classification and regression tasks. This is primarily due to the varying semantic and spatial details captured by the output feature maps of different layers. Though shallow-level feature maps excel at capturing fine edge details, they lack semantic context, whereas deep-level feature maps contain rich semantic information but exhibit coarse spatial resolution. As a result, fully leveraging the advantages of the decoupled head structure becomes challenging due to the mismatched characteristics of these feature maps.

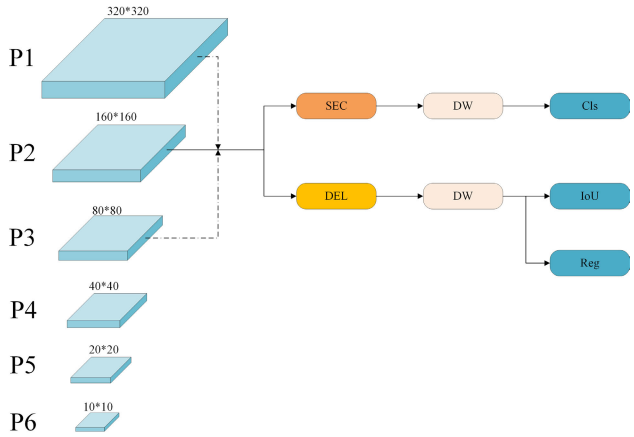

FIGURE 5. The Structure of the Special Decoupled Head (SDH). In this figure, only the {P2, P3, P4, P5} layers are shown, and each layer requires the incorporation of contextual feature information from the upper and lower layers (represented by dashed lines, indicating the connection of SEC and DEL modules with contextual feature information). DW represents 3×3 depth-wise separable convolutions.

To address the above issues, this paper proposes the Special Decoupled Head (SDH). At each output layer, the SDH incorporates the Semantic Encoding for Classification (SEC) and Detail Encoding for Localization (DEL) modules to combine more suitable contextual semantic information for their respective tasks, as illustrated in Fig. 5. In the classification branch, the objective is to determine the category based on the image's feature information, and it is crucial to utilize feature maps with richer semantic information. Therefore, the SEC module, which enriches semantic information while sacrificing some spatial details, is used for the classification task. In the localization branch, where capturing finer edge details is vital for accurate bounding box regression, the DEL module is utilized, which emphasizes edge details at the cost of some semantic information. Experimental results indicate that the proposed SDH detection head effectively resolves the conflict between classification and regression tasks, and it outperforms both traditional decoupled and coupled head structures in addressing this issue. In road scene object detection tasks, the SDH three-layer joint output structure seamlessly resolves the conflict between the focus on classification and regression tasks. It efficiently detects occluded targets and accurately determines their respective categories, thereby enhancing the precision of road scene object detection.

1) THE SEC MODULE

In existing algorithms, the classification task often focuses on determining the features of the object's key and salient parts but ignores the sparsely distributed feature regions of the key parts. This can lead to feature redundancy. This paper believes that during classification, the surrounding contextual features also help to infer the object's category. For example, the presence of cars often coincides with the presence of buses, allowing the recognition of occluded or distant car objects by utilizing features from a larger region. Therefore, this paper introduces the SEC module, which aims to leverage feature maps with richer semantic information for the classification task, as depicted in Fig. 6. In this figure, the output feature map P undergoes a downsampling operation using a convolutional operation with a kernel size of k=3 and a stride of s=2. Then, the downsampled feature map is concatenated with the deep-level feature map, as shown in Equation (1). It is noteworthy that the SEC module not only utilizes the key features from the P layer but also integrates richer semantic information from the deep-level feature maps. Experimental results indicate that this design is effective for inferring the object's category.

$$Out = Concat\left(P+1, CBS\left(P\right)\right). \tag{1}$$

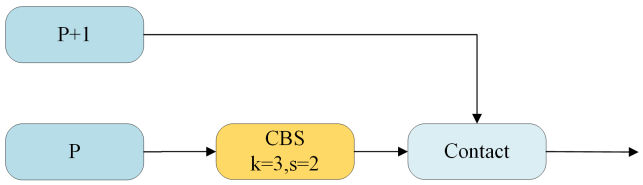

FIGURE 6. The structure of the SEC module.

2) THE DEL MODULE

In existing algorithms, the localization task is usually performed by regressing the corner points of the bounding boxes based on the edge details in the P-layer feature map. However, relying only on the P-layer feature information can cause errors in the predicted bounding boxes. To solve this problem, this paper proposes the DEL module, as shown in Fig. 7. It is believed that deep-level feature maps contain more comprehensive edge detail information, which is crucial for

accurate bounding box regression. Meanwhile, shallow-level feature maps have larger spatial dimensions, allowing the capturing of the entire object and providing more information to determine the object's shape and size. In Fig. 7, the P-layer output incorporates feature information from both the P+1 and P-1 layers. The P+1 layer contributes more edge detail information, while the P-1 layer provides a broader feature map. This integration is performed according to Equation (2). Experimental results indicate that the SDH structure, consisting of the SEC and DEL modules, effectively leverages richer semantic and edge detail information, thus decoupling the classification and localization tasks and improving their performance.

$$Out = CBS\left(CBS\left(P-1\right) + CBS\left(Upsample\left(CBS\left(P\right)\right)\right)\right) + CBS\left(Upsample\left(P+1\right)\right) + CBS\left(P\right). \quad (2)$$

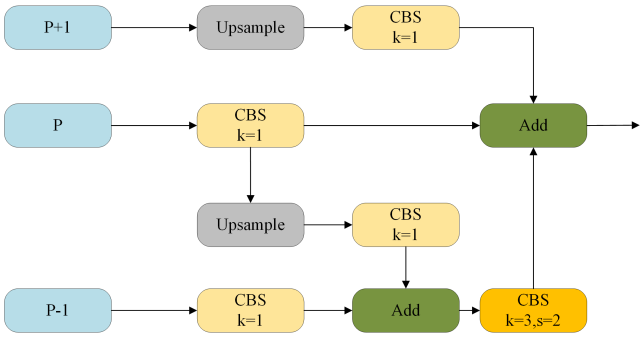

FIGURE 7. The Structure of the DEL Module. In the figure, the upsampling operation (Upsample), 1×1 convolution, and 3×3 convolution are all used to adjust the channel number and dimensions of the feature maps to fit the Add operation.

III. EXPERIMENTS

A. DATASETS

To validate the effectiveness and authenticity of the proposed CMS-YOLO algorithm for autonomous driving applications, experiments were conducted using the publicly available datasets, namely Udacity Self-Driving and BDD100K. The input image size was adjusted to 640×640.

The Udacity Self-Driving dataset is designed by Udacity specifically for autonomous driving algorithm competitions. It provides 2D annotations for consecutive video frames. The dataset consists of 11 classes, including biker, car, pedestrian, trafficLight, trafficLight-Green, trafficLight-GreenLeft, trafficLight-Red, trafficLight-RedLeft, trafficLight-Yellow, trafficLight-YellowLeft, and truck. However, since is a limited number of labels for the trafficLight-YellowLeft class, which results in fluctuations in model performance, this class was not involved in the experiments, and the evaluation was performed using 10 classes. The dataset contains a total of 29,800 images with a resolution of 512×512. These images were randomly split into a training set (26,579 images) and a validation set (3,221 images) at a ratio of 9:1.

BDD100K is one of the widely used datasets for autonomous driving. It consists of 100,000 images in total, and 80,000 images have been annotated. Among the annotated images, 70,000 were used as the training set, while the remaining 10,000 images were used for validation. The BDD100K dataset includes 10 classes: Person, Rider, Car, Bus, Truck, Bike, Motor, Train, TrafficLight, and Traffic Sign. However, due to the lack of instances for the Train class, it was excluded from the experiments. Additionally, considering the need to identify specific traffic light states such as green, red, yellow, or none, the class TrafficLight was further divided into TrafficLight-Green, TrafficLight-Red, TrafficLight-Yellow, and TrafficLight-None. Thus, the final version of the BDD100K dataset used for experiments comprises 12 classes: Person, Rider, Car, Bus, Truck, Bike, Motor, TrafficLight-Green, TrafficLight-Red, TrafficLight-Yellow, TrafficLight-None, and Traffic Sign. Fig. 8 presents some images in the Udacity Self-Driving and BDD100K datasets.

FIGURE 8. Images from the udacity self-driving and BDD100K datasets.

B. EVALUATION METRICS AND EXPERIMENTAL SETUP

The evaluation metrics used in this study include precision (P), recall (R), mean average precision at IoU threshold 0.5 (mAP@0.5), parameter count (Parameters), and frame per second (FPS). The specific calculation formulas are shown below:

$$P = \frac{TP}{TP + FP}. \quad (3)$$

$$R = \frac{TP}{TP + FN}. \quad (4)$$

$$mAP@0.5 = \frac{\sum_{i=1}^{N_{class}} \int_0^1 P_i R_i dR}{N_{class}}. \quad (5)$$

Formulas (3) and (4) show the evaluation metrics used in this study. P represents precision, which denotes the ratio of correctly predicted positive samples among the predicted positive samples. R represents recall, which denotes the ratio of correctly predicted positive samples among all actual positive samples. TP refers to the number of true positive detection boxes with an intersection over union (IoU) greater than the specified threshold. FP refers to the number of false positive detection boxes with an IoU less than or equal to the threshold. FN refers to the number of false negative cases where ground-truth boxes are not detected. In Formula (5), @0.5 denotes the threshold set for IoU at 0.5. N_{class} represents the total number of classes. $\int_0^1 P_i R_i dR$ represents the accuracy of detecting the target in the i-th class, and mAP refers to the mean average precision, which represents the average accuracy across all classes.

The experimental configuration and environment for the study were as follows: The GPU model used was NVIDIA RTX 4090 with 24 GB memory, and the CPU model was Intel i7-13700KF. The software used were PyTorch 1.13.1, Python 3.9, and Cuda 11.7.1. The operating system used was Windows 10. The training parameters were set as follows: the input image size was configured to 640×640 pixels, and the pre-trained weight used was yolov5s.pt. The maximum number of iterations was set to 300, with a batch size of 16 and 8 num_workers for multi-threading. The optimization algorithm chosen was stochastic gradient descent (SGD) with a momentum of 0.937 and a weight decay coefficient of 0.0005. Additionally, the initial learning rate was set to 0.01 and dynamically decayed using the cosine annealing algorithm, reaching a final learning rate of 0.002.

C. RESULTS AND ANALYSIS

1) ABLATION EXPERIMENTS

To verify the effectiveness and authenticity of the proposed CMS-YOLO algorithm, ablation experiments were conducted on the Udacity Self-Driving and BDD100K datasets. These experiments evaluated the impact of various modules and techniques on the overall performance of the algorithm. The experimental results are listed in Table 1 and Table 2, where the symbol "✓" indicates the use of a specific module or technique.

As shown in Table 1, the experimental results on the Udacity Self-Driving dataset demonstrate the significant improvements achieved by each enhancement of the CMS-YOLO algorithm compared to YOLOv5. Specifically, in Scheme 0, using YOLOv5 as the baseline, the algorithm achieves a mAP@0.5 of 86.1% and a mAP@0.5:0.95 of 55.8%. In Scheme 1, replacing the network's backbone with the CD structure leads to a 2.2% improvement in mAP@0.5 and a 1.8% improvement in mAP@0.5:0.95. Scheme 2, which incorporates the MFFPN as the feature fusion module, shows an increase of 2.9% in mAP@0.5 and 3.5% in mAP@0.5:0.95. In Scheme 3, the SDH is introduced by modifying the original coupled head, leading to a 2.8% improvement in mAP@0.5 and a 5.3% improvement in mAP@0.5:0.95. Combining CD with MFFPN in Scheme 4 improves the mAP@0.5 by 3.6% and mAP@0.5:0.95 by 4.9%. Scheme 5 combines CD with SDH, achieving a 3.5% improvement in mAP@0.5 and a 5.8% improvement in mAP@0.5:0.95. Finally, in Scheme 6, the CMS-YOLO algorithm incorporates all three modules together, and it improves mAP@0.5 by 4.2% and mAP@0.5:0.95 by 6.5%. These results demonstrate the effectiveness and performance gains of the CMS-YOLO algorithm across different modules and configurations, verifying its superiority over YOLOv5 on the Udacity Self-Driving dataset.

The experimental results on the BDD100K dataset demonstrate that each enhancement in the CMS-YOLO algorithm brings a significant performance improvement compared to YOLOv5. In Scheme 1, replacing the original backbone with the CD structure leads to a 1.8% improvement in mAP@0.5 and a 2.7% improvement in mAP@0.5:0.95. Scheme 2, which replaces the FPN+PAN structure with MFFPN, achieves a remarkable improvement of 4.7% in mAP@0.5 and 3.5% in mAP@0.5:0.95. In Scheme 3, incorporating the SDH structure as the detection head leads to a substantial improvement of 5.7% in mAP@0.5 and 4.9% in mAP@0.5:0.95. Combining the CD structure with MFFPN in Scheme 4 leads to a significant improvement of 6.4% in mAP@0.5 and 5.0% in mAP@0.5:0.95. Scheme 5, which combines CD with SDH, shows an even greater improvement of 6.8% in mAP@0.5 and 5.6% in mAP@0.5:0.95. Finally, in Scheme 6, the CMS-YOLO algorithm achieves the highest performance by combining all three modules, with a significant improvement of 7.2% in mAP@0.5 and 5.9% in mAP@0.5:0.95. These results fully demonstrate the effectiveness of the CMS-YOLO algorithm, showing its superiority over YOLOv5 in various modules and configurations when evaluated on the challenging BDD100K dataset.

In Tables 3 and 4, regardless of the consideration of target size, CMS-YOLO consistently outperforms other models, achieving higher AP and AR values across various IOU thresholds. When we factor in the target size, CMS-YOLO's superiority becomes even more apparent, with significantly higher AP and AR values compared to YOLOv5s. To elaborate further, CMS-YOLO attains remarkable AP values of 52.8% and 18.8% for small object detection, showcasing improvements of 2.3% and 9.1%, respectively, compared to YOLOv5s. The AR values for CMS-YOLO in small object detection reach impressive figures of 65.5% and 24.1%, demonstrating substantial improvements of 5.4% and 10.3%, respectively. These experimental results confirm the effectiveness of CMS-YOLO in detecting small objects within road scenes.

TABLE 1. Ablation experiments of the CMS-YOLO algorithm on the udacity self-driving dataset.

Scheme	CD	MFFPN	SDH	mAP@0.5(%)	mAP@0.5:0.95(%)	Parameters(Mb)
0				86.1%	55.8%	6.76
1	✓			88.3%	57.6%	9.57
2		✓		89.0%	59.3%	7.10
3			✓	88.9%	61.1%	17.70
4	✓	✓		89.7%	60.7%	10.01
5	✓		✓	89.6%	61.6%	21.37
6	✓	✓	✓	90.3%	62.3%	21.91

TABLE 2. Ablation experiments of CMS-YOLO algorithm on BDD100K dataset.

Scheme	CD	MFFPN	SDH	mAP@0.5(%)	mAP@0.5:0.95(%)	Parameters(Mb)
0				51.9%	25.2%	6.76
1	✓			53.7%	27.9%	9.57
2		✓		56.6%	28.7%	7.10
3			✓	57.6%	30.1%	17.70
4	✓	✓		58.3%	30.2%	10.01
5	✓		✓	58.7%	30.8%	21.37
6	✓	✓	✓	59.1%	31.1%	21.91

TABLE 3. Comparison of object detection performance on different-sized objects between YOLOv5s and CMS-YOLO using the udacity self-driving dataset.

	IoU	Area	maxDets	YOLOv5s	CMS-YOLO
Average Precision (AP)	0.50:0.95	all	100	0.543	0.553
	0.50	all	100	0.847	0.898
	0.75	all	100	0.590	0.617
	0.50:0.95	small	100	0.505	0.528
	0.50:0.95	medium	100	0.693	0.790
	0.50:0.95	large	100	0.759	0.789
Average Recall (AR)	0.50:0.95	all	1	0.372	0.392
	0.50:0.95	all	10	0.620	0.663
	0.50:0.95	all	100	0.629	0.675
	0.50:0.95	small	100	0.601	0.655
	0.50:0.95	medium	100	0.746	0.756
	0.50:0.95	large	100	0.811	0.841

TABLE 4. Comparison of object detection performance on different-sized objects between YOLOv5s and CMS-YOLO using the BDD100K Dataset.

	IoU	Area	maxDets	YOLOv5s	CMS-YOLO
Average Precision (AP)	0.50:0.95	all	100	0.243	0.273
	0.50	all	100	0.498	0.570
	0.75	all	100	0.217	0.229
	0.50:0.95	small	100	0.097	0.188
	0.50:0.95	medium	100	0.340	0.359
	0.50:0.95	large	100	0.561	0.562
Average Recall (AR)	0.50:0.95	all	1	0.202	0.208
	0.50:0.95	all	10	0.375	0.378
	0.50:0.95	all	100	0.404	0.418
	0.50:0.95	small	100	0.138	0.241
	0.50:0.95	medium	100	0.529	0.567
	0.50:0.95	large	100	0.673	0.696

Fig. 9, 10, 11, and 12 show the comparison of mAP@0.5 between CMS-YOLO and YOLOv5s, YOLOv8s for each class on the Udacity Self-Driving and BDD100K datasets, respectively. The results exhibit great performance improvements, particularly in classes like biker, pedestrian, person, and Traffic Sign, where dense small objects are prevalent. The experimental results fully demonstrate the efficacy of the CD structure, which enables the network to capture a larger receptive field and extract more comprehensive and informative features. The MFFPN structure plays a crucial role in fusing shallow and deep features, contributing to enhanced feature representation and improved performance. Besides, the SDH structure effectively resolves the conflict between classification and regression tasks, thereby improving the network's perception and discrimination abilities. The CMS-YOLO algorithm effectively reduces both false positives and false negatives and greatly enhances the detection accuracy of dense small objects in road scenes. It is verified to be highly effective in capturing intricate details and significantly improving detection precision in challenging scenarios.

2) COMPARATIVE EXPERIMENTS

By conducting these comparative experiments on the Udacity Self-Driving and BDD100K datasets, the detection capabilities of the CMS-YOLO algorithm were comprehensively evaluated. The current state-of-the-art algorithms were separately trained on the Udacity self-driving and BDD100K

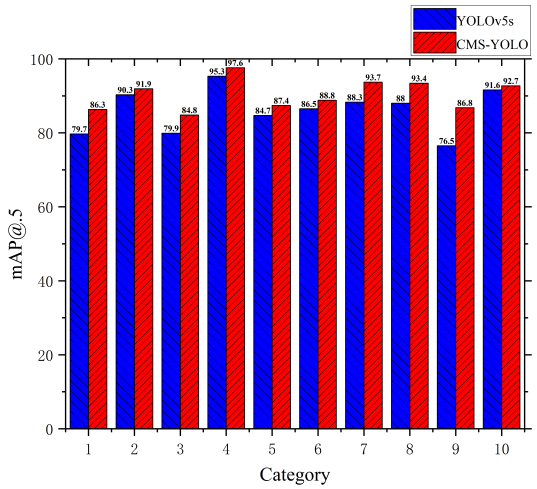

FIGURE 9. Comparison of Results between CMS-YOLO and YOLOv5s on the Udacity self-driving dataset for each class. The class numbers and their corresponding class names are as follows: 1: biker, 2: car, 3: pedestrian, 4: trafficLight, 5: trafficLight-Green, 6: trafficLight-GreenLeft, 7: trafficLight-Red, 8: trafficLight-RedLeft, 9: trafficLight-Yellow, 10: truck.

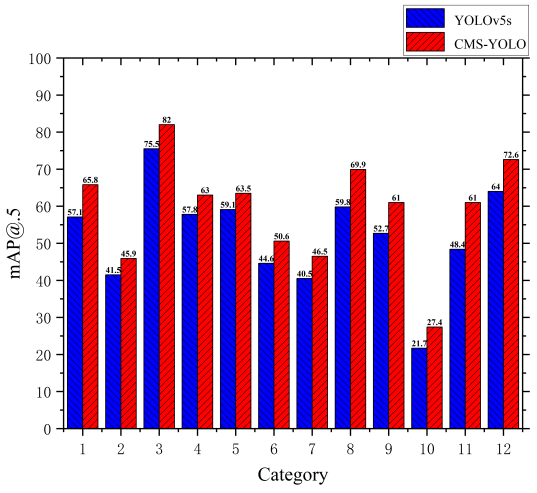

FIGURE 10. Comparison of Results between CMS-YOLO and YOLOv5s on the BDD100K dataset for each class. The class numbers and their corresponding class names are as follows: 1: Person, 2: Rider, 3: Car, 4: Bus, 5: Truck, 6: Bike, 7: Motor, 8: TrafficLight-Green, 9: TrafficLight-Red, 10: TrafficLight-Yellow, 11: TrafficLight-None, 12: Traffic Sign.

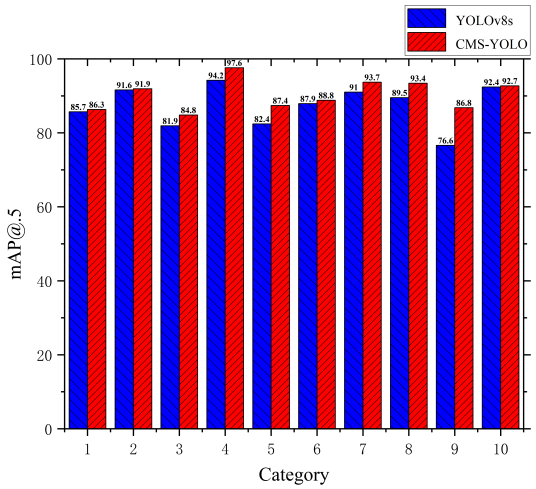

FIGURE 11. Comparison of results between CMS-YOLO and YOLOv8s on the udacity self-driving dataset for each class.

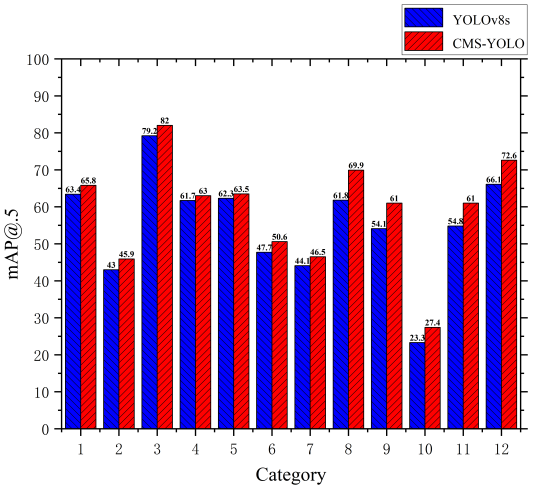

FIGURE 12. Comparison of results between CMS-YOLO and YOLOv8s on the BDD100K dataset for each class.

datasets, and the results were then compared to provide a detailed analysis of their respective performance advantages and disadvantages.

Table 5 presents a comparative analysis of the proposed CMS-YOLO algorithm and current state-of-the-art algorithms on the Udacity Self-Driving dataset. CMS-YOLO outperforms the popular two-stage detection algorithm, Faster-RCNN, with a notable improvement of 11.7% in mAP@0.5 and a substantial increase of 23.8 FPS in detection speed. This signifies a significant improvement in both accuracy and efficiency. When compared to the single-stage detection algorithm, SSD, CMS-YOLO overcomes the challenge of detecting densely packed small objects and achieves a remarkable improvement of 38.2% in mAP@0.5, accompanied by an increase of 12.7 FPS in detection speed. These improvements successfully overcome the limitations of SSD in handling such targets. Compared to YOLOv4, CMS-YOLO demonstrates a noteworthy increase of 6.9% in mAP@0.5, accompanied by a significant enhancement in detection accuracy. Moreover, when compared to the four versions of YOLOv5 (s, m, l, x), CMS-YOLO achieves a substantial improvement of 4.2%, 3.8%, 3.2%, 2.5% in mAP@0.5 and achieves a real-time detection speed of 34.5 FPS. In the field of autonomous driving, this is considered to meet the requirements of real-time detection, as it surpasses the standard threshold of 30 FPS. Importantly, this performance exceeds that of the l and x versions by 13.3 FPS and 25.4 FPS, respectively. Against TPH-YOLOv5, which incorporates a popular transformer structure, CMS-YOLO achieves a commendable 2.4% improve-

ment in mAP@0.5 while maintaining comparable detection speed. In comparison to YOLOv7, CMS-YOLO achieves a commendable improvement of 2.2% in mAP@0.5 and a simultaneous increase in detection speed by 14.5 FPS. Although YOLOv7-tiny demonstrates high real-time detection speed, its detection accuracy lags. In contrast, CMS-YOLO achieves an impressive improvement of 11.5% in mAP@0.5, demonstrating a fine balance between speed and accuracy. Additionally, when compared to YOLOv8s, one of the latest algorithms in the YOLO series, the proposed CMS-YOLO algorithm achieves a superior balance between detection accuracy and speed, leading to an improvement of 3% in mAP@0.5.

TABLE 5. Comparative experiments of the CMS-YOLO algorithm on the udacity self-driving dataset.

Model	Backbone	mAP@0.5(%)	mAP@0.5:0.95(%)	FPS
Faster-RCNN [23]	ResNet-50	78.6%	49.4%	10.7
SSD [24]	VGG-16	52.1%	20.3%	21.8
YOLOv4 [12]	CSP-DarkNet53	83.4%	50.2%	52.3
YOLOv5s	C3	86.1%	55.8%	75.2
YOLOv5m	C3	86.5%	56.5%	40
YOLOv5l	C3	87.1%	58.6%	21.2
YOLOv5x	C3	87.8%	60.1%	9.1
TPH-YOLOv5 [25]	C3-Transformer	87.9%	59.3%	36.3
YOLOv7 [15]	E-ELEN	88.1%	60.2%	20
YOLOv7-tiny [15]	E-ELEN	78.8%	39.8%	278
YOLOv8s	C2F	87.3%	59.2%	87.1
CMS-YOLO	**CD**	**90.3**%	**62.3**%	**34.5**

Table 6 provides a comprehensive comparison between the proposed CMS-YOLO algorithm and the current state-of-the-art algorithms on the BDD100K dataset. CMS-YOLO surpasses Faster-RCNN with a significant improvement of 16% in mAP@0.5. In contrast to YOLOv3, CMS-YOLO achieves an impressive increase of 19% in mAP@0.5, demonstrating its remarkable performance. Besides, CMS-YOLO exhibits a noteworthy improvement of 7.9% in mAP@0.5 when compared to YOLOv4. Similarly, when compared to the four versions of YOLOv5 (s, m, l, x), CMS-YOLO exhibits a substantial increase of 7.2%, 7%, 4.5%, 2.8% in mAP@0.5, highlighting its enhanced detection capability. In comparison to TPH-YOLOv5, CMS-YOLO shows a commendable improvement of 2.5% in mAP@0.5, demonstrating its superiority. When measured against IMP YOLOv5 and MCS-YOLO, CMS-YOLO outperforms both with a 7.9% and 5.5% improvement in mAP@0.5, respectively. Additionally, in contrast to YOLOv7, CMS-YOLO achieves a commendable improvement of 2.3% in mAP@0.5, demonstrating its remarkable performance. While there may be variances in real-time detection speed, CMS-YOLO outperforms YOLOv7-tiny with an impressive improvement of 8% in mAP@0.5, showing its enhanced detection accuracy. Furthermore, when compared to YOLOv8s, the proposed CMS-YOLO algorithm achieves a notable improvement of 4% in mAP@0.5. The experimental results fully verify the superiority of CMS-YOLO to current state-of-the-art algorithms: it achieves higher detection accuracy while effectively fulfilling the demands of real-time detection scenarios.

TABLE 6. Comparative experiments of the CMS-YOLO algorithm on the BDD100K dataset.

Model	Backbone	mAP@0.5(%)	mAP@0.5:0.95(%)	FPS
Faster-RCNN [23]	ResNet-50	43.1%	19.8%	10.7
YOLOv3 [11]	DarkNet53	40.1%	16.5%	23.8
YOLOv4 [12]	CSP-DarkNet53	51.2%	23.2%	52.3
YOLOv5s	C3	51.9%	25.2%	75.2
YOLOv5m	C3	52.1%	25.8%	40
YOLOv5l	C3	54.6%	27.2%	21.2
YOLOv5x	C3	56.3%	28.5%	9.1
TPH-YOLOv5 [25]	C3-Transformer	56.6%	28.2%	36.3
IMP YOLOv5 [26]	G-C3	51.2%	27.6%	35
MCS-YOLO [27]	C3-ST	53.6%	28.6%	55
YOLOv7 [15]	E-ELEN	56.8%	29.0%	20
YOLOv7-tiny [15]	E-ELEN	51.1%	21.7%	278
YOLOv8s	C2F	55.1%	27.9%	87.1
CMS-YOLO	**CD**	**59.1**%	**31.1**%	**34.5**

3) DETECTION RESULT VISUALIZATION

To provide a more intuitive representation of the superiority of the proposed CMS-YOLO algorithm, six images were randomly selected from the Udacity Self-Driving and BDD100K datasets for comparison. The comparative results are given in Fig. 13 and 14, showcasing the algorithm's performance in detecting objects in real-world road scenes.

FIGURE 13. Visualization Comparison on the udacity self-driving dataset.

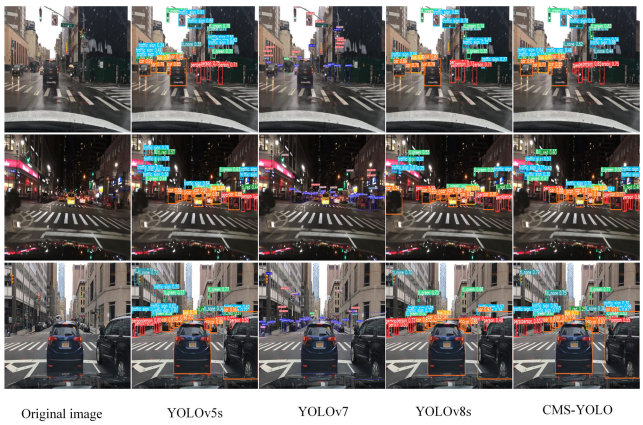

FIGURE 14. Visualization comparison on the BDD100K dataset.

Fig. 13 and 14 present the visual comparison results of CMS-YOLO, YOLOv5s, YOLOv7, and YOLOv8s on

the Udacity Self-Driving and BDD100K datasets. It can be seen that YOLOv5s, YOLOv7 and YOLOv8s have several instances of missed detections or false detections for objects such as traffic signals, traffic signs, and pedestrians. In contrast, CMS-YOLO can accurately detect these objects. Besides, the CMS-YOLO algorithm demonstrates higher confidence and robustness in detecting dense small objects in road scenes under varying weather and lighting conditions. This remarkable performance advantage makes CMS-YOLO well-suited for diverse autonomous driving scenarios.

IV. CONCLUSION

This paper addresses the issue of real-time and high-precision object detection in autonomous driving scenarios by proposing the CMS-YOLO algorithm, which aims to achieve a balance between accuracy and efficiency. To accomplish this, several key improvements have been introduced. Firstly, the backbone of the original network is replaced with the CD module to enable the network to extract features with a larger receptive field and richer information. This prepares the network for subsequent feature fusion. Secondly, the MFFPN is designed to efficiently fuse shallow and deep-level features, thereby avoiding feature information loss during propagation. Lastly, the SDH is introduced to resolve the conflict between classification and regression tasks, which further enhances the algorithm's performance. Experimental results demonstrate the effectiveness of the CMS-YOLO algorithm, and it achieves a mAP@0.5 of 90.3% and 59.1% on the Udacity Self-Driving and BDD100K datasets, respectively, showing an improvement of 4.2% and 7.2% over the baseline algorithms. Meanwhile, the proposed algorithm achieves a real-time detection speed of 34.5 FPS, meeting the requirements of accuracy and real-time performance in autonomous driving scenarios. Compared to current mainstream algorithms, the CMS-YOLO algorithm achieves superior performance and is highly suitable for autonomous driving applications. Besides, it has great potential for practical implementation in real-world autonomous driving systems.

REFERENCES

[1] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4224–4231, Sep. 2018.

[2] H. Gao, Y. Qin, C. Hu, Y. Liu, and K. Li, "An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, pp. 1–12, 2021.

[3] J. Ni, K. Shen, Y. Chen, and S. X. Yang, "An improved SSD-like deep network-based object detection method for indoor scenes," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–15, 2023.

[4] S. He, L. Chen, S. Zhang, Z. Guo, P. Sun, H. Liu, and H. Liu, "Automatic recognition of traffic signs based on visual inspection," *IEEE Access*, vol. 9, pp. 43253–43261, 2021, doi: 10.1109/ACCESS.2021.3059052.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[6] Q. Lin, R. Wang, Y. Wang, F. Zhou, and N. Guo, "Target detection algorithm incorporating visual expansion mechanism and path syndication," *IEEE Access*, vol. 11, pp. 56973–56982, 2023.

[7] 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.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[13] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[14] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[15] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[16] Y. Chen, W. Zheng, Y. Zhao, T. H. Song, and H. Shin, "DW-YOLO: An efficient object detector for drones and self-driving vehicles," *Arabian J. Sci. Eng.*, vol. 48, no. 2, pp. 1427–1436, Feb. 2023.

[17] H. Wang and W. Zang, "Research on object detection method in driving scenario based on improved YOLOv4," in *Proc. IEEE 6th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, vol. 6, Mar. 2022, pp. 1751–1754.

[18] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "YOLOv4–5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021, doi: 10.1109/TIM.2021.3065438.

[19] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 636–644.

[20] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[22] Z. Chen, C. Yang, Q. Li, F. Zhao, Z.-J. Zha, and F. Wu, "Disentangle your dense object detector," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 4939–4948, doi: 10.1145/3474085.3475351.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[24] W. Liu, D. Anguelov, and D. Erhan, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[25] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.

[26] Q. Luo, J. Wang, M. Gao, Z. He, Y. Yang, and H. Zhou, "Multiple mechanisms to strengthen the ability of YOLOv5s for real-time identification of vehicle type," *Electronics*, vol. 11, no. 16, p. 2586, Aug. 2022.

[27] Y. Cao, C. Li, Y. Peng, and H. Ru, "MCS-YOLO: A multiscale object detection method for autonomous driving road environment recognition," *IEEE Access*, vol. 11, pp. 22342–22354, 2023.

ZHENYANG LV received the B.S. degree from Suzhou City University, Suzhou, China, in 2020. He is currently pursuing the M.Eng. degree with the College of Information Engineering, Yancheng Institute of Technology, Yancheng, China. His current research interests include computer vision technology and image processing technology.

RUGANG WANG received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 1999, the M.S. degree from Jinan University, Guangzhou, China, in 2007, and the Ph.D. degree from Nanjing University, Nanjing, China, in 2012. He is currently a Professor with the College of Information Engineering, Yancheng Institute of Technology, Yancheng, China. His current research interests include optical communication networks, novel and key devices for optical communication systems, and image processing technology.

YUANYUAN WANG, photograph and biography not available at the time of publication.

FENG ZHOU received the B.S. and M.S. degrees from Southeast University, Nanjing, China, in 2004 and 2012, respectively. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA. Since 2017, he has been an Associate Professor with the College of Information Engineering, Yancheng Institute of Technology, Yancheng, China. His current research interests include cooperative communication, computer vision technology, and image processing technology.

NAIHONG GUO, photograph and biography not available at the time of publication.