Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.030940

ARTICLE

Improved Convolutional Neural Network for Traffic Scene Segmentation

Fuliang Xu, Yong Luo, Chuanlong Sun and Hong Zhao*

College of Mechanical and Electrical Engineering, Qingdao University, Qingdao, 266071, China

*Corresponding Author: Hong Zhao. Email: zhaohong@qdu.edu.cn

Received: 04 May 2023 Accepted: 01 August 2023 Published: 15 December 2023

ABSTRACT

In actual traffic scenarios, precise recognition of traffic participants, such as vehicles and pedestrians, is crucial for intelligent transportation. This study proposes an improved algorithm built on Mask-RCNN to enhance the ability of autonomous driving systems to recognize traffic participants. The algorithm incorporates long and short-term memory networks and the fused attention module (GSAM, GCT, and Spatial Attention Module) to enhance the algorithm's capability to process both global and local information. Additionally, to increase the network's initial operation stability, the original network activation function was replaced with Gaussian error linear unit. Experiments were conducted using the publicly available Cityscapes dataset. Comparing the test results, it was observed that the revised algorithm outperformed the original algorithm in terms of AP_{50}, AP_{75}, and other metrics by 8.7% and 9.6% for target detection and 12.5% and 13.3% for segmentation.

KEYWORDS

Instance segmentation; deep learning; convolutional neural network; attention mechanism

1 Introduction

Intelligent transportation and autonomous driving have significantly advanced with the rapid growth of information technology [1]. Intelligent transportation has emerged as a prominent trend in the automotive industry [2], with a key component being the observation and understanding of the driving environment by self-driving cars [3]. Numerous scholars have developed new algorithms to aid self-driving cars in better understanding their environment. The current mainstream algorithms can be divided into single-stage and two-stage algorithms depending on the processing method [4].

Two-stage instance segmentation algorithms, such as SDS [5], DeepMask [6], Mask-RCNN [7], and SGN [8], have become typical examples in this field. The two-stage algorithm originated in 2014 when Pinheiro et al. [5] created the SDS algorithm. Although the SDS algorithm fell short in terms of processing speed and recognition accuracy compared to current algorithms, it provided a framework for further research. In 2015, Dai et al. [9] published an MNC model using cascaded structure-sharing convolutional features. Then, in 2017, He et al. [10] proposed the Mask-RCNN model based on the Faster-RCNN model, marking a new phase in instance segmentation task with Mask-RCNN serving as the benchmark model.

Two-stage algorithms built on area candidate networks tend to achieve higher accuracy. However, models such as Mask-RCNN and Cascade Mask R-CNN [11] require a lot of memory resources and have long inference times when trained on small batches. On the other hand, single-stage algorithms such as YOLACT [12], Blend Mask [13], and CondInst [14], which perform both localization and segmentation, are quicker and more suitable for the real-time demands of autonomous driving, but they lack accuracy. Additionally, single-stage instance segmentation techniques such as the YOLACT series [12] , SOLO [15] series, and CenterMask [16] can fully utilize positional data in images, resulting in high accuracy and segmentation speed, but their training time is usually lengthy. Although single-stage algorithms typically outperform two-stage algorithms in terms of speed, they often lack the accuracy and precision achieved by two-stage algorithms.

Researchers have explored various techniques to increase accuracy. Zhang et al. [17] used random undersampling (RUS) for detection and classification. Lin et al. [18] produced a number of target candidate regions through a region recommendation approach network. Then, to enhance parameter learning, they trained a convolutional neural network (CNN) using a library of real inspection picture samples. Yuan et al. [19] introduced a spatial attention mechanism that focuses on regions requiring attention to increase the segmentation accuracy. On the other hand, Ya et al. [20] proposed a two-stage coarse-to-fine search technique called structure-to-modular NAS (SM-NAS) to discover GPU-friendly designs with improved modular-level architectures for object identification and efficient module combinations. This technique involved a structure-level search phase that seeks effective ways to combine various modules, followed by a modular-level search phase that evolves each module and drives the Pareto front end toward a quicker task-specific network with a significant increase in inference speed.

Deep learning has greatly advanced computer vision, with CNN becoming the architecture of choice for popular models. However, recurrent neural networks (RNNs) offer distinct advantages over CNNs in capturing target context, and RNN/LSTM applications are currently gaining popularity in the recognition sector. For example, Xiang et al. [21] used deep learning techniques, principally based on a blend of long short-term memory (LSTM) and CNN, for quick target extraction from films. Yang et al. [22] integrated a standard target detector with an LSTM module to further enhance the detection performance of intelligent driving. In response to recent developments in computer vision, Mallick et al. [23] substituted the gated recurrent units (GRU) for LSTM as a decoder in image captioning models. The combination of the attention mechanism and the GRU decoder increased the accuracy.

In this study, Mask-RCNN is used as a benchmark for enhancing and optimizing the algorithm. Based on recent research findings, the superiority of two-stage algorithms over single-stage algorithms is believed to be valid. By examining a few of the research papers on LSTM-based recognition, it is evident that most of them use more sophisticated hybrid models. However, overly intricate networks can hinder recognition. GRU offers faster training compared to LSTM due to its reduced tensor operations. Therefore, this study investigates an improved algorithm that combines Mask-RCNN with GRU. In summary, it seeks to enhance the capabilities of the existing algorithm.

The following are the contributions of this study:

1. To make the algorithm useful for complex traffic circumstances, it is optimized using Mask-RCNN as the benchmark. The incorporation of the CBAM attention mechanism has improved the segmentation algorithm's accuracy. However, the CBAM module is computationally intensive and prioritizes accuracy over speed, which makes it unsuitable for safe driving in smart cars. To address these challenges, the CBAM model is enhanced to make it more suitable for complicated traffic scenarios by leveraging the aforementioned modularity and spatial attention. To improve feature extraction, the huge convolution is converted to a tiny convolution, and the channel attention module

is replaced with the less complicated GCT attention mechanism. These modifications reduced the model's complexity.

2. The introduction of network-end long and short-term memory methods improves the feature extraction performance of the algorithm. This approach enables the algorithm to prioritize the interplay between global and local information and effectively connect recent information with older information to increase overall performance.

3. To increase the algorithm's robustness, the Gaussian error linear unit (GELU) is used as the algorithm's activation function.

This study is divided into five sections. Section 1 introduces the development history of the research; Section 2 describes the improvement made to the network; Section 3 describes the experimental setup and configuration environment; Section 4 presents the experimental results and visually showcases the performance of the improved algorithm; Section 5 summarizes the findings and contributions of this study.

2 Network Improvement

During training, models often need to acquire and analyze a large amount of data, but frequently only a small subset of that data is relevant at any given time. The attention mechanism imitates the capacity of the human brain to identify critical regions in complex visual images. This mechanism has proven to be effective in achieving tasks such as target identification, semantic segmentation, and instance segmentation.

2.1 Converged Attention Module (GSAM)

The convolutional block attention module (CBAM) [24] accurately assigns and processes information, leading to improved target detection accuracy. In this study, a modification inspired by the CBAM structure called Gaussian context transformer (GCT) [25] and spatial attention module (SAM), collectively referred to as GSAM, is obtained. Fig. 1 shows the structure of GSAM.

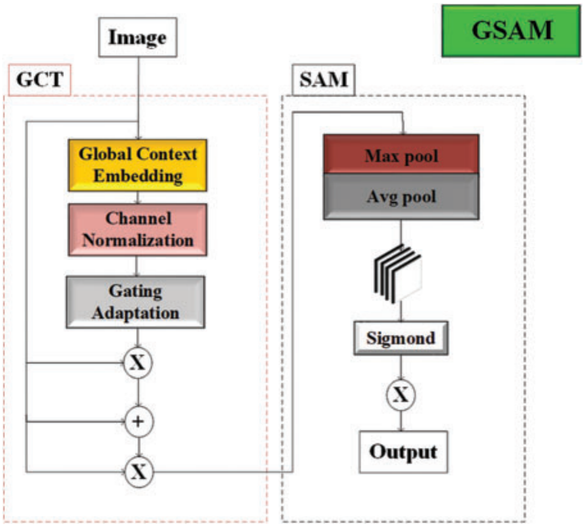


Figure 1: GSAM structure diagram

GSAM consists of GCT and SAM, integrated within the bottleneck of the ResNet-50 base module. The experimental trials conducted demonstrate the complementary nature of this structure and how the two can work together to enhance the precision and stability of feature extraction. To improve feature extraction for complex traffic scenarios, the complexity is reduced by replacing the big convolution with the tiny convolution.

The GCT consists of three operations: global context aggregation (GCA), channel normalization, and Gaussian context excitation (GCE).

(1) Global Context Aggregation (GCA)

$$\mathrm{s}_c = \alpha_c \, ||\chi_c||_2 = \alpha_c \left\{ \left[\sum_{i=1}^{H} \sum_{j=1}^{W} \left(x_c^{ij} \right)^2 \right] + \ell \right\}, \tag{1}$$

where α_c is the adaptive embedding output weight; χ_c is the input feature map; H and W are the spatial dimensions; ℓ is a constant. Large receptive fields help avoid local confusion. Therefore, this global inline embedded module aggregates global context information for each channel.

(2) Channel Normalization

$$\hat{s}_c = \frac{\sqrt{C} s_c}{||x_c||_2} = \frac{\sqrt{C} s_c}{\left[\left(\sum_{C=1}^{C} S_c^2 \right) + \ell \right]^{\frac{1}{2}}}, \tag{2}$$

where C is the number of channels. Normalization can construct competing relationships between neurons using limited computational resources, similar to local response normalization, which GCT uses to perform cross-channel feature normalization, i.e., channel normalization, defined as expressed in Eq. (2).

(3) Gaussian Context Excitation (GCE)

$$\hat{x}_c = x_c \left[1 + \tanh \left(\gamma_c \hat{s}_c + \beta_c \right) \right], \tag{3}$$

where γ_c is the gating weight and β_c is the bias. The GCT introduces a gating mechanism to those mentioned earlier, as expressed in Eq. (3), which further promotes the competition or synergistic relationships between neurons.

Based on the experimental analysis, it is observed that adding GCT before the convolution operation yields better results. Therefore, the CBAM improved by GCT, i.e., the GSAM, is inserted between each lifting and lowering dimension operation to increase the resolution of image information through the convolutional neural network.

The spatial attention component of GSAM remains unchanged and follows the original spatial attention module approach. First, it aggregates the channel information of the feature map through two pooling operations to generate two 2D mappings, as expressed in Eqs. (4) and (5).

$$F_{avg}^{s} \in R^{1\times H\times W}, \tag{4}$$

$$F_{\max}^{s} \in R^{1\times H\times W}. \tag{5}$$

These mappings are concatenated and convolved through a standard convolution layer to produce a 2D spatial attention map. Then, the map is normalized using a sigmoid function to obtain the final attention map, as expressed in Eq. (6).

$$\begin{aligned} M_s(F) &= \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma\left(f^{7\times7}\left(\left[F^s_{avg}; F^s_{\max}\right]\right)\right). \end{aligned} \tag{6}$$

Fig. 2 shows the improved identity and convolution block structures of the ResNet-50 network.

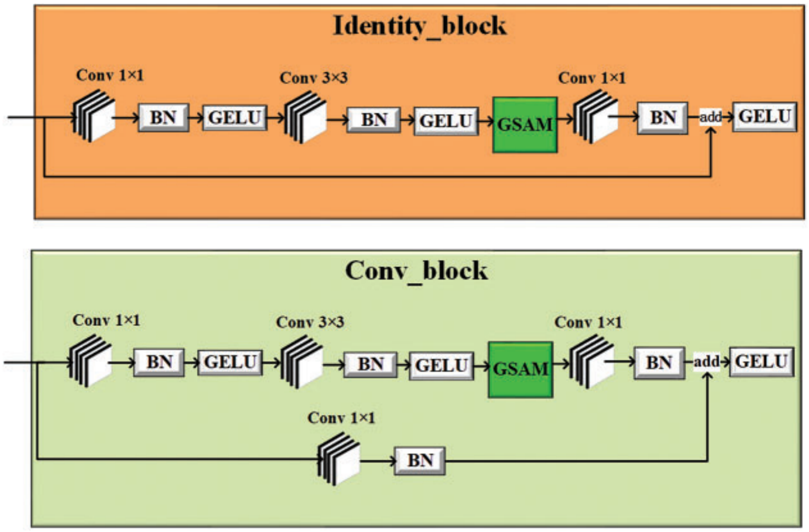


Figure 2: Improved identity block and convolution block

The improved backbone layer increases the ability to detect small targets among images while maintaining computational speed [26]. Additionally, the system's robustness is improved due to the inclusion of the GCT network within the CBAM, and this enhancement is particularly significant for the ResNet network.

2.2 Activation Functions

In a biological sense, a neuron is activated only when the weighted sum of the signals transmitted by the primary dendrites exceeds a specific threshold value. Inspired by this biological operation mechanism, activation functions were introduced [27].

The activation function activates some neurons within the neural network during runtime and transmits activation information to the subsequent neural network layer. By introducing nonlinear factors, activation functions enable neural networks to solve nonlinear problems, enhancing the expressiveness of nonlinear models [28]. Activation functions preserve and map the "characteristics of activated neurons" to the next layer through mathematical functions.

Mask-RCNN uses rectified linear unit (ReLU) as the activation function, as expressed in Eq. (7).

$$f(x) = \max(0, x). \tag{7}$$

Although the ReLU function is computationally straightforward, it is not differentiable at the zero point and has no output when the input value is negative [29]. As a result, the entire neural network "dies" if any neuron encounters a negative input value.

Additionally, when the input value is negative, all negative values instantly become zero, making it challenging to train or fit the model accurately to the data. As a result, any negative input to the ReLU activation function will cause the graph's values to instantly change to zero. This sudden change to zero values due to negative inputs leads to improper mapping, which affects the overall outcomes [30].

To address these limitations, the ReLU function is replaced with the GELU function, which offers improved smoothness and strengthens the network's stability. The GELU function is mathematically expressed as follows:

$$GELU(x) = x * \Phi(x). \tag{8}$$

The GELU function is differentiable at the zero point. It yields output even when the input value is negative, preventing the initial death of neurons and ensuring the required model smoothing. Additionally, the GELU function introduces stochastic regularization, which establishes a stochastic connection between the input and output and prevents the disappearance of the gradient. This significantly reduces the likelihood of neuron necrosis and improves the algorithm's stability.

2.3 Gated Circulation Unit

The GRU is a gate mechanism unit in RNN, similar to LSTM, but can achieve the same effect compared to LSTM. Compared to LSTM, GRU is easier to train and can significantly improve training efficiency. Therefore, GRU is chosen for this experiment. Fig. 3 shows the GRUs structure.

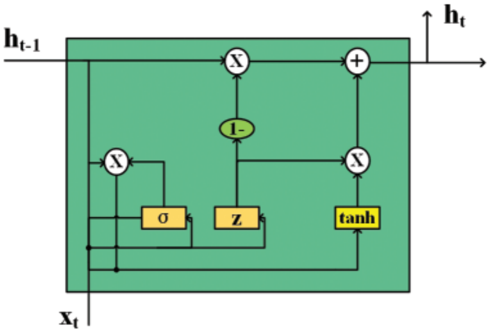


Figure 3: GRU network structure

The most important innovation that distinguishes GRU from LSTM is the design of an "update memory" phase [31], which involves two steps: forgetting and remembering. The expression for updating memory is as follows:

$$h^t = (1 - z) \odot h^{t-1} + z \odot h'. \tag{9}$$

In Eq. (9), z is used as a gating signal, and its signal range is closer to 1 for more data to be remembered and closer to 0 for more data to be forgotten. The $(1-z)h^{t-1}$ represents the selective forgetting of the previous state, and the $(1-z)$ is used as a forgetting gate to control the "forgetting" of some unimportant information in the h^{t-1} dimension. Similarly, $z \odot h'$ represents the selective memory of the current state, where z is a memory gate controlling the retention of some critical information in the h^t dimension.

This updated memory mechanism keeps the amount of data "constant", with a certain amount of information about the previous state "forgotten" at (1−z) and a certain amount of information about the current state "remembered" at z. The combination of forgetting (1−z) and remembering z forms a linkage mechanism that keeps the amount of data constant and prevents the risk of gradient explosion [32].

The number of GRU layers is set to two. Using two layers of GRU can compress the data into a highly "condensed" form. Additional layers beyond two will only lead to information loss during the information compression process and gradient disappearance during training. Fig. 4 shows the structure of the ResNet-50-GRU network.

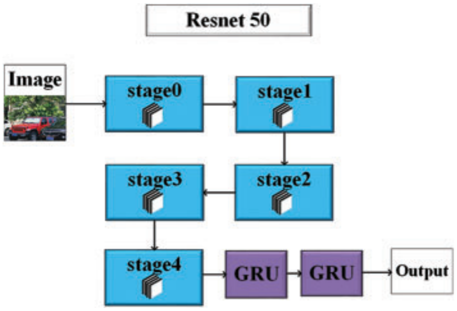


Figure 4: ResNet-50-GRU network

3 Experimental Environment and Evaluation Index

The Cityscapes dataset, released in 2015 by Mercedes-Benz, contained 5000 finely labeled images of driving scenes in urban environments, including pedestrians, cyclists, vehicles, and other essential urban landscape targets. Although this dataset is recognized as one of machine vision's most authoritative and professional image segmentation datasets, it lacks some exceptional cases, such as snow and rain scenes. To address this, additional datasets were created to compensate for this deficiency.

All algorithms used in this experiment were executed on PyCharm 2021.3.1 IDE with Windows 10 hardware platform, i7 CPU, and 8 GB NVIDIA RTX 2070 SUPER GPU. The maximum epoch was set to 100, using the SGD optimizer with pre-training parameters and Warmup method, and a batch size of 4. The Pytorch library version used was 1.14.0.

In this study, average precision (AP) was used as a metric to measure the performance of model instance segmentation in conjunction with practical considerations. The metrics include AP_{50}, AP_{75}, AP_s, AP_m, and AP_l, where AP_{50} and AP_{75} represent different IOU thresholds of 0.5 and 0.75, respectively. AP_s, AP_m, and AP_l represent the average accuracy of target objects with small, medium, and large areas, respectively. This comprehensive criterion setting allows for a thorough evaluation of the model's segmentation performance.

4 Experimental Results

4.1 Data Analysis

To verify the effectiveness of the added network modules, the experimental ablation metrics of detection and segmentation on the Cityscapes dataset are presented in Tables 1 and 2, respectively.

The original Mask-RCNN algorithm is denoted as Base, while the network with the CBAM attention mechanism added is referred to as Base-CBAM. Similarly, the networks with GSAM and GRU added are labeled as Base-GSAM, Base-GRU, and Base-GSAM-GRU. PointRend and BlendMask are used to split the network for other instances.

Table 1: Comparison of target detection results

Model	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Base	0.548	0.327	0.112	0.335	0.561
Base-GRU	0.552	0.332	0.111	0.348	0.557
Base-CBAM	0.554	0.346	0.110	0.362	0.578
Base-GSAM	0.559	0.347	0.106	0.361	0.584
PointRend	0.563	0.354	0.106	0.368	0.597
Base-CBAM-GRU	0.580	0.363	0.116	0.380	0.611
BlendMask	0.590	0.365	0.118	0.386	0.623
Base-GSAM-GRU	**0.596**	**0.368**	**0.121**	**0.391**	**0.627**

Table 2: Comparison of instance segmentation results

Model	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Base	0.487	0.233	0.055	0.257	0.516
Base-GRU	0.492	0.237	0.056	0.260	0.528
Base-CBAM	0.503	0.246	0.049	0.271	0.532
Base-GSAM	0.523	0.262	0.054	0.285	0.546
PointRend	0.525	0.260	0.053	0.286	0.548
Base-CBAM-GRU	0.527	0.262	0.053	0.286	0.556
BlendMask	0.532	0.262	0.056	0.289	0.560
Base-GSAM-GRU	**0.534**	**0.264**	**0.057**	**0.290**	**0.569**

From the comparative analysis of Tables 1 and 2, it can be concluded that the GRU, CBAM, and GSAM modules improve the algorithm's performance, with their performance improvement degree as GSAM > CBAM > GRU. The inclusion of the CBAM attention mechanism led to increased accuracy in the segmentation algorithms. To strengthen the CBAM model and make it more suitable for complex traffic conditions, the attention mechanism was developed using the aforementioned modularity and spatial attention. The model's complexity was slightly reduced through improved feature extraction, smalling large convolutions, and replacing the channel attention module with a simpler GCT attention mechanism. The addition of the GSAM module and GRU module enabled the network to effectively combine global and local information, leading to improved identification of the vehicle environment and the traffic participants, such as vehicles, pedestrians, and cyclists, resulting in enhanced detection and segmentation abilities and overall generalization. After replacing its activation function, the initial death problem of network neurons was partially resolved, and the robustness of the network improved. Furthermore, its segmentation accuracy was found to be advantageous compared with other popular segmentation networks used nowadays. Figs. 5 and 6 show the change process of mAP, loss, and learning rate of the Base-GSAM-GRU network automatically generated by the algorithm during the training process, respectively. It was observed that the training starts to converge at 20 epochs.

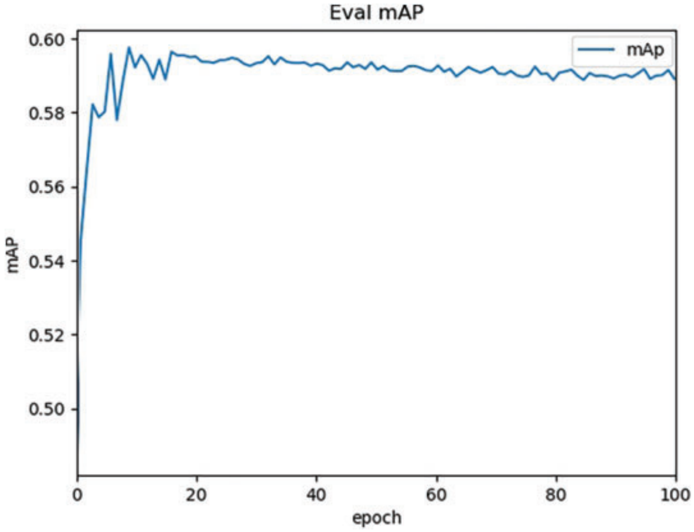


Figure 5: mAP change plot

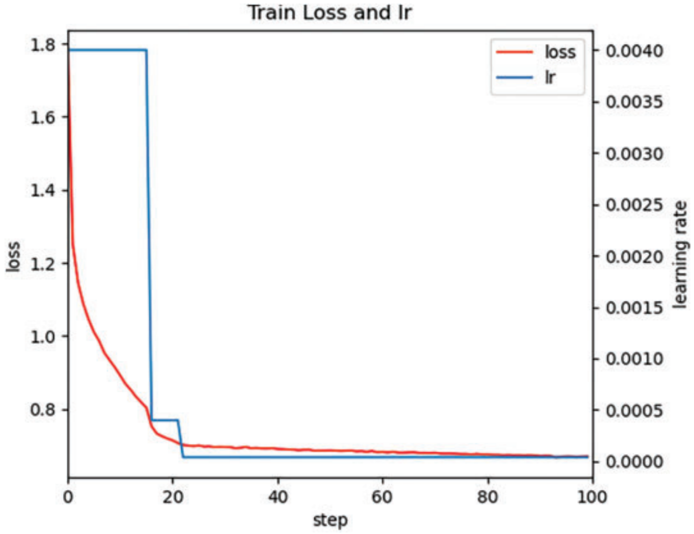


Figure 6: Loss and learning rate change diagram

Based on the above evaluation indexes, such as AP, further evaluation of the algorithm's performance was conducted using detection accuracy P and recall rate R to verify its accuracy further. The mathematical formulae are expressed in Eqs. (10) and (11).

$$p = \frac{TP}{TP + FP}, \tag{10}$$

$$R = \frac{TP}{TP + FN}. \tag{11}$$

For the network training, models with iteration periods of 30, 50, 70, 90, and 100 were selected for evaluation. Table 3 presents the network training loss and detection accuracy.

Table 3: Network training and loss accuracy

Training period	Loss	Accuracy
30	1.726	71.5%
50	1.636	74.6%
70	1.504	79.5%
90	1.424	83.4%
100	1.374	84.7%

As the number of training cycles of the network increases, the network loss effectively decreases and tends to improve, indicating that the accuracy of this advanced network model is constantly improving. The network model with a training period of 100 was selected, and the test samples were randomly selected from the dataset test set. Fig. 7 shows the tested and calculated detection accuracy and the target detection categories. The average detection accuracy of the experiment reached 88%.

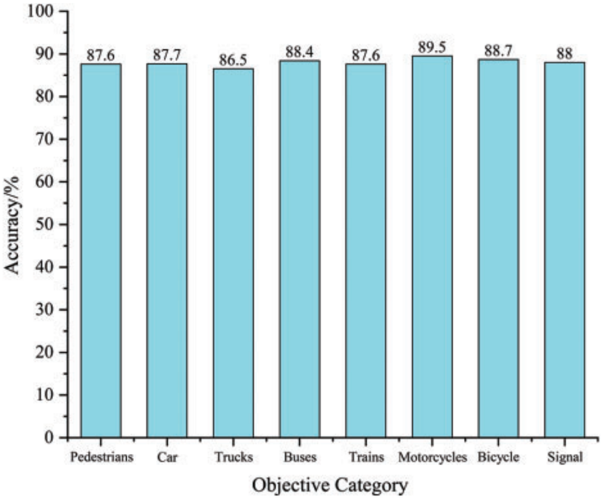
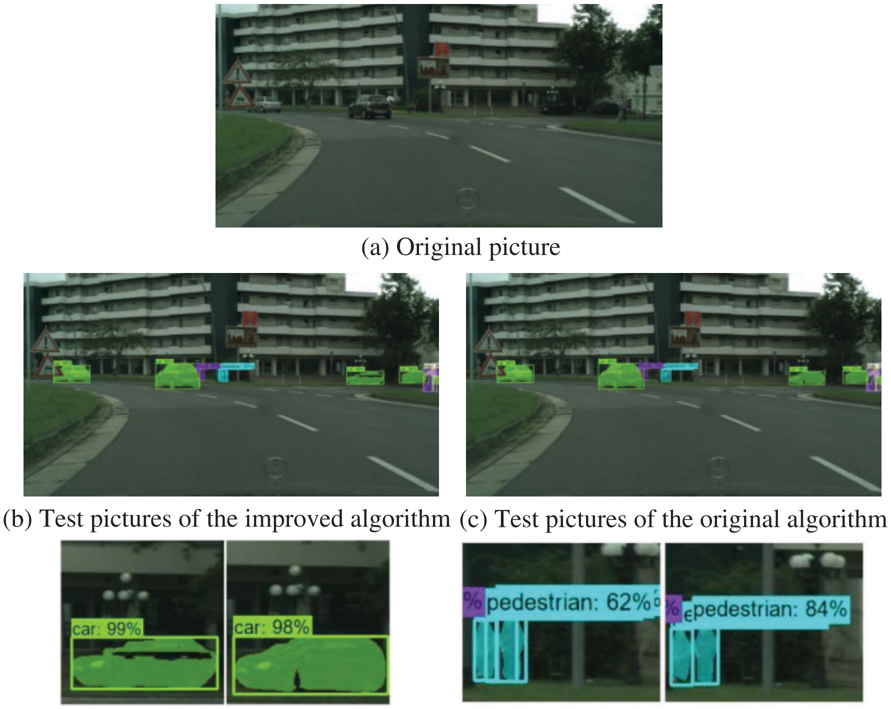


Figure 7: Target detection accuracy

4.2 Visual Verification

Fig. 8 shows the comparison of the improved algorithm with the original algorithm. It was observed that the improved algorithm is more accurate in pedestrian recognition and occlusion handling.

Comparing the details of Figs. 8b and 8e, it was observed that the right side of each figure represents the original algorithm, while the left side represents the improved algorithm. The improved algorithm exhibits more accurate target segmentation compared to the original algorithm. For instance, the original algorithm misidentifies two pedestrians as three, whereas the improved algorithm accurately handles the occlusion problem and does not mistake street lights that do not belong to the car as a car.



(a) Original picture

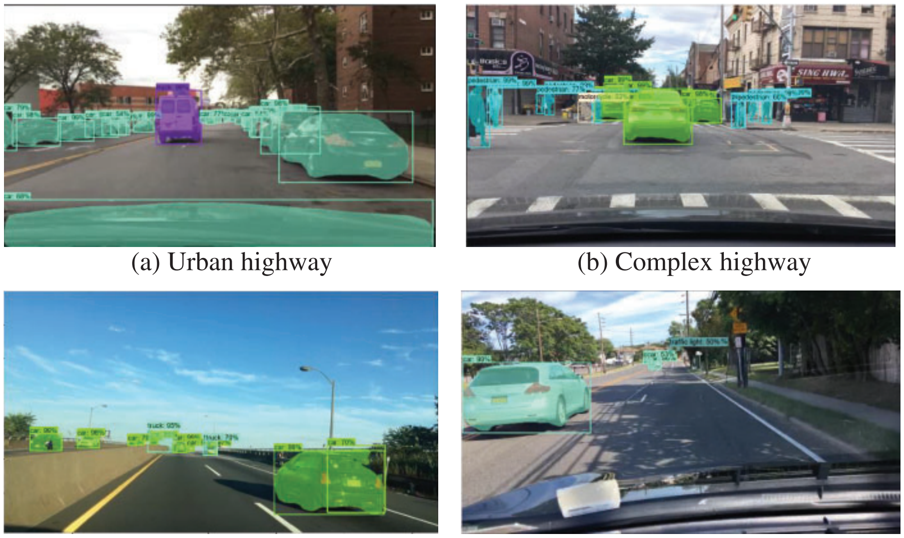
(b) Test pictures of the improved algorithm (c) Test pictures of the original algorithm

(d) Comparison picture of a car

(e) Comparison picture of pedestrian

Figure 8: Comparison picture

To demonstrate the network's instance segmentation effectiveness under various traffic scenes, a network model with 100 training iterations was selected for testing, and the test results are shown in Figs. 9–12.



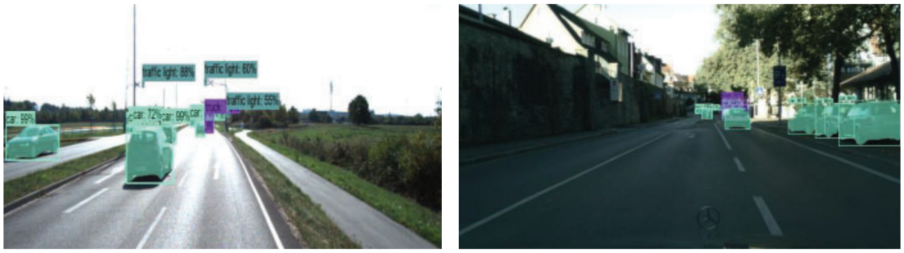
(a) Urban highway

(b) Complex highway

(c) Expressway

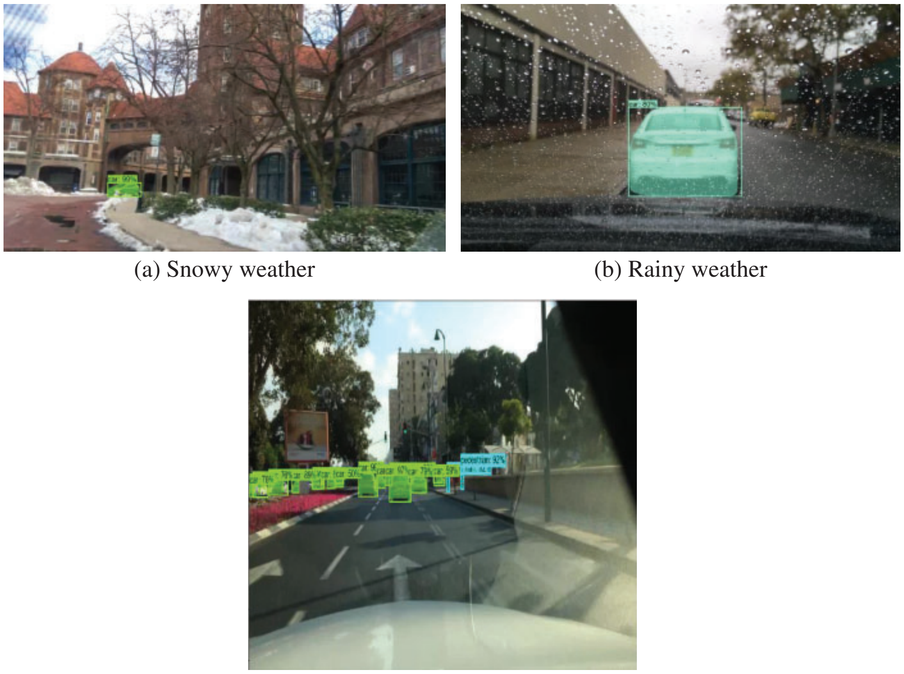
(d) Suburban highway

Figure 9: Detection results in natural scenes



(a) Light occlusion (b) Shadow occlusion

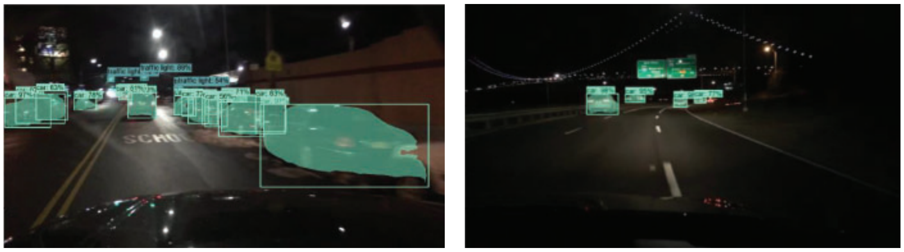
Figure 10: Detection results in light scenes



(a) Snowy weather (b) Rainy weather

(c) Long-distance scene

Figure 11: Detection results in fuzzy scenes



(a) One-way street scene (b) Two-way scene

Figure 12: Detection results in dark scenes

Fig. 9 shows the segmentation results under natural traffic scenes, with Figs. 9a–9d depicting urban highway, complex highway, expressway, and suburban highway, respectively. The improved algorithm effectively performed target recognition segmentation in the above scenes, with detection results consistent with the actual outcomes and appropriate calibration frame range.

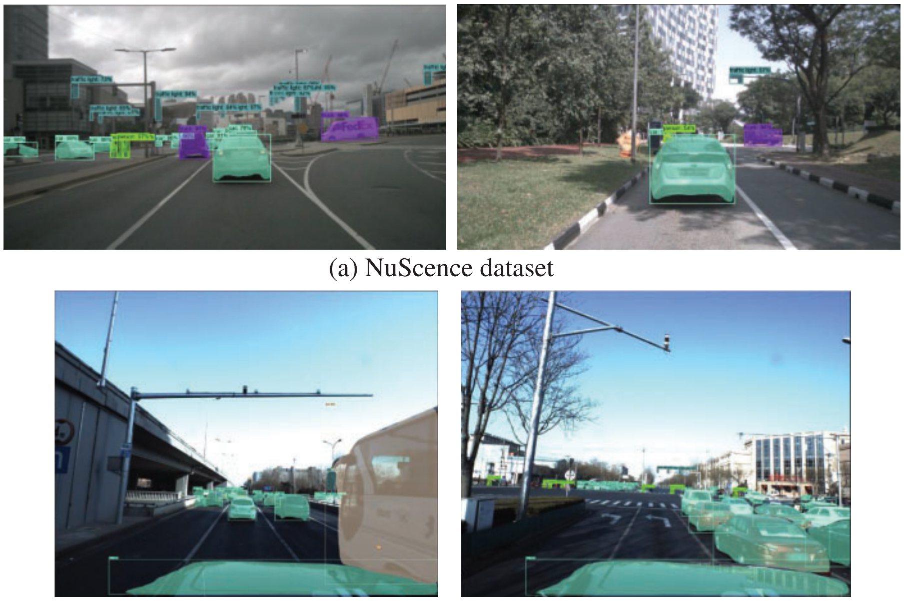
Fig. 10 shows the segmentation results under light traffic scenes, with Fig. 10a depicting the light-obscured scene and Fig. 10b depicting the shadow-obscured scene. Changes in light brightness significantly impact the target appearance, and its color and texture shape are disturbed, causing an increase in segmentation difficulty. However, the segmentation results show that the improved algorithm successfully handles the detection and segmentation tasks in such scenarios.

Fig. 11 shows the segmentation results under blurred scenes, withFigs. 11a–11c depicting snowy weather, rainy weather, and distant scenes, respectively. In these scenes, the shape and appearance of the target are affected, resulting in blurred imaging. However, the segmentation results show that the improved network achieves accurate detection and segmentation in these scenes.

Fig. 12 shows the experimental results under dark traffic scenes, with Fig. 12a depicting a single-lane scene and Fig. 12b depicting a two-lane scene. In dark scenes, changes in target shape, color, and texture are more pronounced compared to blurred scenes. However, the experimental results show that the improved algorithm still has good feasibility and accuracy even in these low-discrimination scenes.

4.3 Migration Experiment Validation

To further verify the recognition and segmentation ability of the network model trained by the algorithm, migration experiments were conducted using the NuScence and Apollo Scape datasets. The images with relevant category ranges were selected from the dataset, and then randomly selected test images were used for testing. Fig. 13 shows the test results of the migration experiments.



(a) NuScence dataset

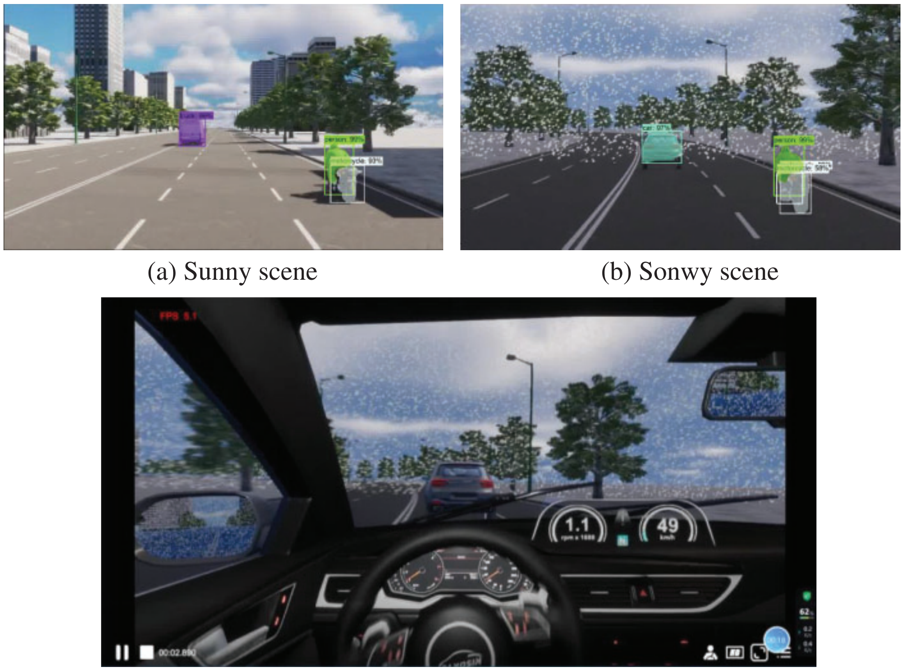
(b) Apollo Scape dataset

Figure 13: Migration experiment test results

4.4 Simulation Environment Validation

To verify the algorithm's effectiveness, the same network model with a training period of 100 was selected for simulation experiments in panosim. The simulation environment included two scenarios:

a sunny day straight road and a snowy day right angle left turn. In the snowy scenario, particle density was set to 20,000 m^{-3}, and the landing speed was 0.8 m/s. The experiment used a monocular camera sensor, and the acquired data was transmitted to the bus and then to the specified address through the specified module for image segmentation by the algorithm. The temperature in a sunny scene is 30 degrees Celsius, the humidity is 60%, the light intensity is 10000 lux, and the incidence angle is 50 degrees. The segmentation results in Fig. 14 show that the improved algorithm effectively segments objects in the simulated scenes.



(a) Sunny scene

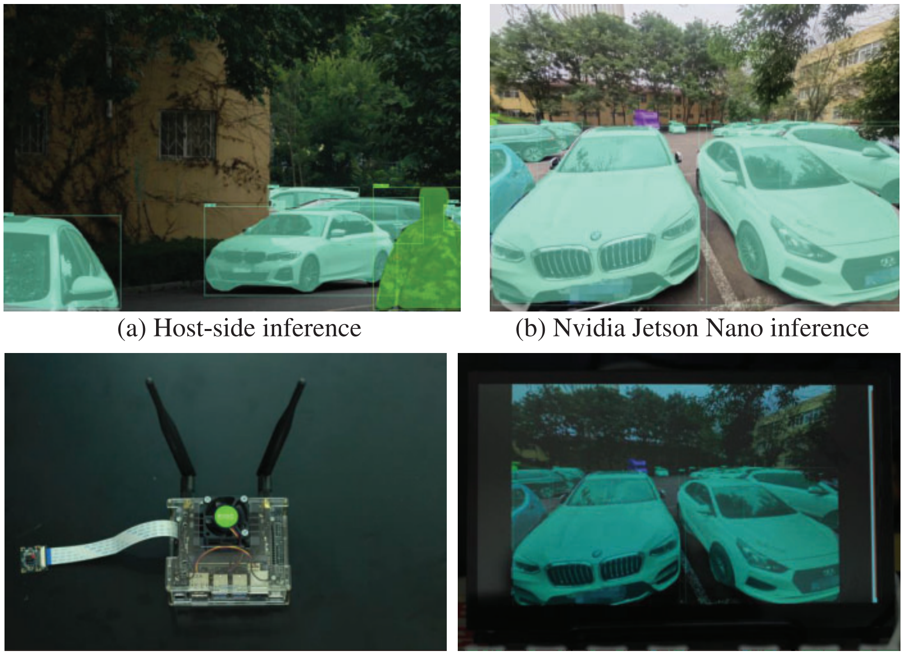
(b) Sonwy scene

(c) Frame diagram of the simulation process

Figure 14: Simulation process and result diagram

To evaluate the algorithm's performance on resource-constrained devices, we conducted optimized inference on the Jetson Nano development board. This board features a 128-core Maxwell GPU, a 4-core ARM A57 CPU clocked at 1.43 GHz, and 4 GB of 64-bit LPDDR4 memory. Our system runs on an ARM version of Ubuntu 18.04.6 LTS, with the model's runtime environment configured with JetPack 4.6.4, Python 3.6.9, CUDA 10.2.300, cuDNN 8.2.1.32, OpenCV 4.1.1 with CUDA, and TensorRT 8.2.1.8.

The segmentation results in Fig. 15 depict the successful segmentation of objects in real-world scenes achieved by our improved algorithm. However, due to the limited computational power of the Nano deployment board, the individual photo inference speed is relatively slow, as shown in Table 4.



(a) Host-side inference

(b) Nvidia Jetson Nano inference

(c) Nvidia Jetson Nano

Figure 15: The diagram of edge deployment

Table 4: Network training and loss accuracy

Model	Time/ms	Device
Base	1848	Nvidia Jetson Nano
Base-GSAM-GRU	1954	
Base	1116	i7 CPU, and 8 GB NVIDIA RTX 2070 SUPER GPU
Base-GSAM-GRU	1176	

5 Conclusion

This study addressed the problem of information loss in instance segmentation networks. An enhanced instance segmentation algorithm based on Mask-RCNN was proposed, introducing a GSAM framework. This framework effectively combined local and global information, mitigating information loss and local ambiguity of the network during convolution. The channel and spatial attention modules within the framework work together, providing complementary functions. Additionally, the inclusion of the CBAM attention mechanism improved the accuracy of the segmentation algorithms. By leveraging the aforementioned modularity and spatial attention, the attention mechanism was enhanced to make the CBAM model more suitable for challenging traffic scenarios. The complexity of the model was partially reduced through enhanced feature extraction, smaller big convolutions, and the replacement of the channel attention module with a simpler GCT attention mechanism. Additionally, the introduction of the GRU structure effectively used deep-level information and enhanced the network's learning capacity. When combined with GRU, the original method exhibited excellent generalization performance, with significant improvements observed across all metrics based on experimental findings.

The experiment's algorithm was executed in Python, which affected its running efficiency a little below average. However, using C/C++ to write the application process will increase operating efficiency. It is important to note that the experimental data and equipment imposed limitations on addressing the recognition of numerous categories in car driving scenes. The follow-up still has room for improvement. Additionally, the method functions well on the server and can be made lighter in the future, such as using the Pruning algorithm for deployment on the edge.

Acknowledgement: The authors thank TopEdit (www.topeditsci.com) for its linguistic assistance during the preparation of this manuscript.

Funding Statement: This work was supported by the National Natural Science Foundation of China (52175236) and Qingdao People's Livelihood Science and Technology Plan (19-6-1-88-nsh).

Author Contributions: Study conception and design: Fuliang Xu; data collection: Chuanlong Sun, Yong Luo; analysis and interpretation of results: Fuliang Xu, Yong Luo; draft manuscript preparation: Fuliang Xu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The experimental dataset is a public dataset. The results of the data experiment are the common property of the research group, and I have no right to disclose it.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Mu, L., Zhao, H., Li, Y., Liu, X. T., Qiu, J. Z. et al. (2021). Traffic flow statistics method based on deep learning and multi-feature fusion. *Computer Modeling in Engineering & Sciences, 129(2),* 465–483. https://doi.org/10.32604/cmes.2021.017276
2. Zhu, H., Sun, C., Zheng, Q., Zhao, Q. (2023). Deep learning based automatic charging identification and positioning method for electric vehicle. *Computer Modeling in Engineering & Sciences, 136(3),* 3265–3283. https://doi.org/10.32604/cmes.2023.025777
3. Zhao, H., Mu, L., Li, Y., Qiu, J. Z., Sun, C. L. et al. (2021). Unregulated emissions from natural gas taxi based on IVE model. *Atmosphere, 12(4),* 478.
4. Sun, C. L., Zhao, H., Mu, L., Xu, F. L., Lu, L. W. (2023). Image semantic segmentation for autonomous driving based on improved U-Net. *Computer Modeling in Engineering & Sciences, 136(1),* 787–801. https://doi.org/10.32604/cmes.2023.025119
5. Pinheiro, P. O., Collobert, R., Dollar, P. (2015). Learning to segment object candidates. *29th Annual Conference on Neural Information Processing Systems (NIPS)*, vol. 8. Montreal, Canada.
6. Thiruvathukal, G. K., Lu, Y. H. (2022). Efficient computer vision for embedded systems. *Computer, 55(4),* 15–19.
7. Girshick, R. (2015). Fast R-CNN. *IEEE International Conference on Computer Vision*, pp. 1440–1448. Santiago, Chile.
8. Liu, S., Jia, J. Y., Fidler, S., Urtasun, R. (2017). SGN: Sequential grouping networks for instance segmentation. *16th IEEE International Conference on Computer Vision (ICCV)*, pp. 3516–3524. Venice, Italy.
9. Dai, J. F., He, K. M., Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3150–3158. Seattle, WA.

10. He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. Venice, Italy.
11. Dong, X. J., Shao, H. L., Li, Z. X., Luo, J. (2022). Intelligent detection of defects in aerospace composite materials based on convolutional neural network. *Aerospace Shanghai, 39(4),* 1554–160 (in Chinese).
12. Bolya, D., Zhou, C., Xiao, F. Y., Lee, Y. J. (2019). YOLACT real-time instance segmentation. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9156–9165. Seoul, South Korea.
13. Chen, H., Sun, K., Tian, Z., Shen, C., Haung, Y. et al. (2020). BlendMask: Top-down meets bottom-up for instance segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8573–8581. Seattle, WA, USA.
14. Tian, Z., Zhang, B. W., Chen, H., Shen, C. H. (2023). Instance and panoptic segmentation using conditional convolutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1),* 669–680.
15. Wang, X. L., Zhang, R. F., Shen, C. H., Kong, T., Li, L. (2022). SOLO: A simple framework for instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44,* 8587–8601.
16. Xuan, Z. M., Ding, J. W., Mao, J. (2022). Intelligent identification method of insulator defects based on CenterMask. *IEEE Access, 10,* 59772–59781.
17. Zhang, D., O'Conner, N. E., Simpson, A. J., Cao, C. J., Little, S. et al. (2022). Coastal fisheries resource monitoring through A deep learning-based underwater video analysis. *Estuarine, Coastal and Shelf Science, 269,* 107815. https://doi.org/10.1016/j.ecss.2022.107815
18. Lin, G., Wang, B., Peng, H., Wang, X., Chen, S. et al. (2019). Multi-target detection and location of transmission line inspection image based on improved faster-RCNN. *Electric Power Automation Equipment, 39(5),* 213–218.
19. Yuan, L., Qiu, Z. (2021). Mask-RCNN with spatial attention for pedestrian segmentation in cyber-physical systems. *Computer Communications, 180,* 109–114.
20. Ya, L. W., Xu, H., Zhang, W., Liang, X. D., Li, Z. G. (2020). SM-NAS: Structural-to-modular neural architecture search for object detection. *34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial*, pp. 12661–12668. New York, USA.
21. Xiang, Y. F., Yan, W. Q. (2021). Fast-moving coin recognition using deep learning. *Multimedia Tools and Applications, 80(16),* 24111–24120.
22. Yang, Y. F. (2022). Application of LSTM neural network technology embedded in English intelligent translation. *Computational Intelligence and Neuroscience, 2022,* 1085577.
23. Mallick, V. R., Naik, D. (2021). Describing image with attention based GRU. *2021 6th International Conference for Convergence in Technology (I2CT)*, Maharashtra, India, Electr Network.
24. Woo, S. H., Park, J., Lee, J. Y., Kweon, I. S. (2018). CBAM: Convolutional block attention module. *15th European Conference on Computer Vision (ECCV)*, pp. 3–19. Munich, Germany.
25. Yang, Z., Zhu, L., Wu, Y., Yang, Y. (2020). Gated channel transformation for visual recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11791–11800. Seattle, WA, USA.
26. Huang, W., Huang, Y., Yao, Y., Yan, Y. (2022). Automatic classification of retinopathy with attention ConvNeXt. *Optics and Precision Engineering, 30,* 2147–2154.
27. Afonso, M., Fonteijn, H., Fiorentin, F. S., Lensink, D., Mooij, M. et al. (2020). Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in Plant Science, 11,* 571299.
28. Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F. et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Doha, Qatar.

29. Xu, H. F., van Genabith, J., Xiong, D. Y., Liu, Q. H., Zhang, J. Y. et al. (2020). Learning source phrase representations for neural machine translation. *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 386–396.
30. Flores, A., Tito-Chura, H., Yana-Mamani, V. (2021). An ensemble GRU approach for wind speed forecasting with data augmentation. *International Journal of Advanced Computer Science and Applications, 12(6),* 569–574.
31. Hu, N., Zhang, D. F., Xie, K., Liang, W., Diao, C. Y. et al. (2022). Multi-range bidirectional mask graph convolution based GRU networks for traffic prediction. *Journal of Systems Architecture, 133,* 102775.
32. Guo, Z. Y., Yang, C. Y., Wang, D. S., Liu, H. B. (2023). A novel deep learning model integrating CNN and GRU to predict particulate matter concentrations. *Process Safety and Environmental Protection, 173,* 604–613.