International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An innovative approach to enhance the safety of Elevator using steel cable damage detection model based on YOLO

Muhammad Wahab Hanif1*, Zhanli Li[1], Rehmat Bashir[2]

Submitted: 11/03/2024 Revised: 26/04/2024 Accepted: 03/05/2024 DOI: 10.1039/b000000x

Abstract: To address the issues such as limited detection device resources and prolonged detection times in surface damage detection of steel cables installed commercial, public, and industrial buildings, advanced deep learning techniques, and Convolutional Neural Networks (CNN) have been investigated in this study and a new network model has been designed. This work proposes a steel cable defect detection network model based on YOLO, incorporating GhostNet into the backbone network, and introducing a novel feature extraction module (ShuffleNC3) based on ShuffleNet and attention mechanisms. Pruning improvements are then applied to the Head part. Experimental results indicate that the improved network achieves approximately1.149% increase in average precision compared to the baseline YOLOv5s. This modification achieves a simultaneous reduction of network computational costs and maintains high recognition accuracy, meeting better requirements for surface damage detection in steel cables. The parameters and computational costs are reduced by approximately 43 % and 31.4%, respectively, while the model size also decreases by 42%.

Keywords: *Deep learning; Convolutional Neural Network; YOLOv5; Steel cables; Attention mechanism; Surface damage detection; Object detection.*

1. Introduction

The steel cable is a bundle of multiple strands of steel wires twisted around a fiber core or a steel wire rope core. In practical use, issues such as damage can lead to economic losses or casualties. Therefore, timely detection and recognition of damage to the steel cable are crucial. Target detection methods are divided into single-stage and two-stage, with YOLO and R-CNN being representative algorithms, respectively. YOLO, known for its superior performance in speed and detecting smaller targets, has been chosen in this study for conducting detection and recognition experiments on damaged steel cable surfaces.

The YOLO object detection algorithm is a widely used single-stage detection algorithm with various versions such as YOLOv3[1], YOLOv5[2], YOLOv7[3]. YOLOv5, considered the most outstanding due to its performance, builds on YOLOv4[4] with some improvements.

Most algorithms and literature in object detection are based on or refer to YOLOv5. Therefore, this study adopts YOLOv5 for research and improvement. The YOLOv5 network can be divided into four main parts: Input, Backbone, Neck Network, and Head Detection Output. The Input part processes the input images, standardizing their size and normalizing operations. Mosaic operations enhance the input data by randomly scaling, splicing, and cropping four input images. This enriches the detection dataset, increases network robustness, reduces Mini-batch values, and lessens the GPU burden. Adaptive anchor box calculations and adaptive image scaling improve subsequent object detection performance.

The Backbone mainly consists of Conv modules, C3 modules, and SPPF modules, aiming to extract image features from input pictures for subsequent object detection work. The C3 module increases network depth and receptive field, enhancing feature extraction performance. The SPPF module, composed of max-pooling and regular convolution, achieves feature extraction at different scales, generating three-scale feature maps to improve detection accuracy.

The Neck Network combines FPN[5] (Feature Pyramid Network) and PANet[6]. It fuses features through top-down and bottom-up networks and combines the Backbone with the Head Detection Output, further enhancing detection capabilities. The Head Detection Output predicts targets of different sizes on the feature map[7]. Like YOLOv4, YOLOv5 uses multi-scale detection heads, showing good detection performance on feature maps of different sizes[8].

In practical applications of steel cable target detection, issues such as limited detection device resources and prolonged detection times may arise. Therefore, this study optimizes YOLO, reducing the model size and parameter numbers while maintaining high detection performance. By reducing model complexity and computational costs, the aim is to alleviate the practical application problems.

2. Experimental methods and improvements

The network in this paper replaces the convolution modules in the Backbone with GhostConv modules, which have higher performance and fewer parameters. Based on the C3 module, ShuffleNet and attention mechanisms are combined with the original C3 module, resulting in the proposed ShuffleNC3 module. This modification enhances the network's detection performance and provides a lighter structure. Finally, improvements are made to the Head Detection module.

Compared to other fifth-generation networks, YOLOv5s has a lighter structure and lower computational cost. To achieve the balance between better detection performance and lower computational

[1]*Computer Science Department, Xian University of Science and Technology, Xian, China. ORCID ID : 0009-0000-0915-9921*
[1]*Computer Science Department, Xian University of Science and Technology, Xian, China. Lizl@xust.edu.cn*
[3] *Mech. Eng., University of Engineering and Technology, Lahore, Pakistan ORCID ID : rehmatbashir@uet.edu.pk*
** Corresponding Author Email: mdw2016@hotmail.com*

costs, this paper proposes improvements based on the YOLOv5s network. The YOLOv5 network structure is shown in Fig 1. The following sections provide a detailed introduction to the GhostConv module and the ShuffleNC3 module proposed in this paper.

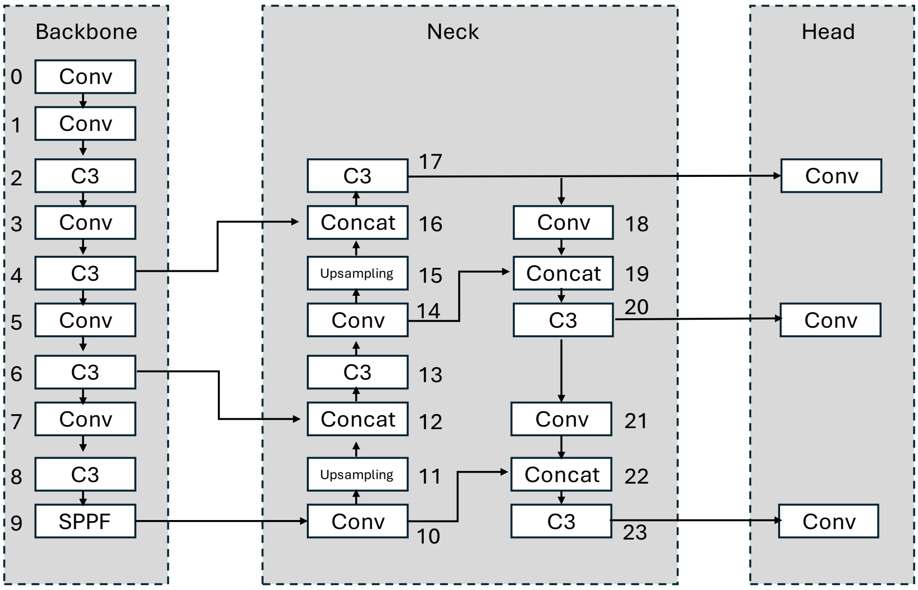

Fig 1. YOLOv5 Network Structure

2.1. GhostConv Convolution module

During feature extraction, neural networks generate a significant amount of similar redundant features, which require significant computational resources[9]. To tackle this issue the GhostNetConv module has been introduced inspiration from GhostNet[10]. This module generates more feature maps with inexpensive operations, thereby reducing memory consumption during the intermediate expansion process. The GhostConv module structure is an improved version of the ordinary convolution module. It transforms the conventional convolution operation into two steps. In the first step, it performs a regular convolution on the input information to obtain some feature maps. In the second step, it performs a linear operation on the feature maps obtained in the first step, generating redundant feature maps. Finally, the outputs of the two steps are concatenated. It is compared with ordinary convolution in Fig 2.

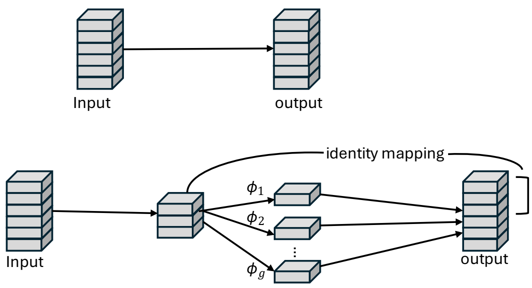

Fig 2. Regular convolution Compared with GhostConv

The GhostConv module uses grouped convolution for linear transformation. Assuming the input feature map size is [C1, H, W]. The output feature map size is [C2, H', W'], number of channels defined by C, and height and width defined by the H and W of the feature map, after grouped convolution with g groups, the input feature map is divided into W×H×C1/g per group. A single convolution kernel becomes K×K×C2/g in size, where K is the kernel size. This significantly reduces the parameter and computation volume of the network. Similarly, assuming the input feature map size is [C1, H, W] and the output feature map size is [C2, H', W'], with conventional convolution kernel size K and linear transformation convolution kernel size D, and g representing the total number of mappings produced for each channel. Assuming K=D and g are much smaller than C1, the following formulas represent the ratio of computation volume and parameter volume between conventional convolution and the GhostConv module:

$$
\begin{aligned}
r_s &= \frac{C_2 \times H' \times W' \times C_1 \times K \times K}{\frac{C_2}{g} \times H' \times W' \times C_1 \times K \times K + (g-1) \times \frac{C_2}{g} \times H' \times W' \times D \times D} \\
&= \frac{C_1 \times K \times K}{\frac{1}{g} \times C_1 \times K \times K + \frac{(g-1)}{g} \times D \times D} \approx \frac{C_1 \times g}{C_1 + g - 1} \approx g
\end{aligned} \tag{1}
$$

Ratio of parameters:

$$
\begin{aligned}
r_c &= \frac{C_1 \times C_2 \times K \times K}{\frac{C_2}{g} \times C_1 \times K \times K + (g-1) \times \frac{C_2}{g} \times D \times D} \\
&\approx \frac{C_1 \times g}{C_1 + g - 1} \approx g
\end{aligned} \tag{2}
$$

Therefore, theoretically, using the GhostConv module can save g times the computational cost and reduce g times the parameters. By introducing the GhostConv module into the Backbone and replacing the ordinary convolution, with g set to 2, it can significantly reduce the computational cost while ensuring network stability.

2.2. ShuffleNC3 Module

2.2.1. ShuffleNetV2

In ShuffleNetV1[11], grouped convolution was used to decrease the number of parameters. In ShuffleNetV2[12], channel split method was proposed on its basis. Channel split replaces grouped convolution, reducing the computational cost required for grouped convolution. It splits the C-dimensional input channels into two branches, obtaining left and right branches. Since the more branches designed in the network and the higher the fragmentation, the slower the network speed, one branch maintains an identity mapping, and the other branch performs feature extraction through three convolutions. After that, a concatenation operation is applied to the outputs. After the Concatenation, since each regular convolution operation is performed on the same branch each time, leading to no information exchange within each group, the idea of channel shuffle is proposed. As shown in Fig 3, for the g groups of feature matrices obtained through channel split, it further divides them into g groups, transforms the position of each channel, and concatenates to create a new feature matrix.

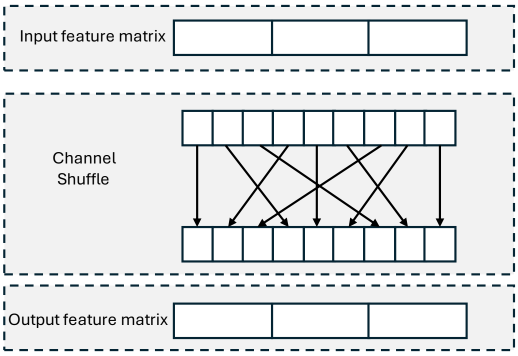

Fig 3. Channel shuffle thought structure

This aims to achieve the fusion of feature information, thereby improving feature reusability and feature extraction performance. The C3 module is an important component in the Backbone, consisting of three convolution modules and one Bottleneck module. In this paper, ShuffleNetV2 is combined with the C3 module, replacing the Bottleneck structure with the ShuffleNet Bottleneck structure. The ShuffleNet Bottleneck arrangement is shown in Fig 4, where the convolution module includes the Conv module, BN normalization module, and activation function module.

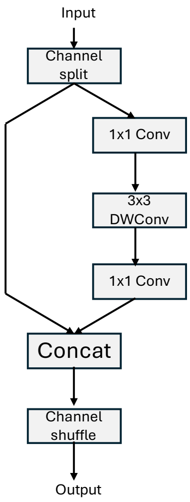

Fig 4. ShuffleNet Bottleneck structure

2.2.2. CBAM Attention

CBAM[13] is a lightweight convolutional attention module. Unlike common attention modules such as SENet[14] and ECANet[15], it performs attention operations not only in the channel dimension but also in the spatial dimension. Its structure is shown in Fig 5.

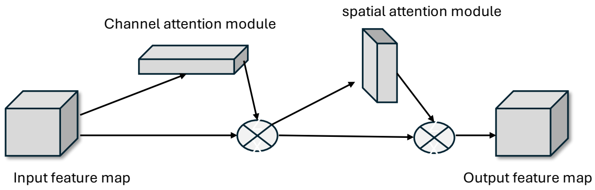

Fig 5. CBAM structure

These two feature maps are fed into a fully connected layer, the output results are added together, and after calculating through the Sigmoid activation function, channel weight coefficients representing the importance of features are obtained. Finally, the original input feature map is multiplied by the channel weight coefficients to acquire an output feature map of size [C, H, W]. The spatial attention module mainly focuses on the positional information of the target. It keeps the dimension unchanged in space and compresses it in the channel, using the same principle as the channel attention module.

In this paper, CBAM attention is added to the ShuffleNet Bottleneck structure. The CBAM attention mechanism enables the network to focus on more relevant areas. Through parallel max-pooling and average-pooling layers, the network, during the feature extraction process, extracts richer and more comprehensive high-level features, thereby improving feature extraction performance.

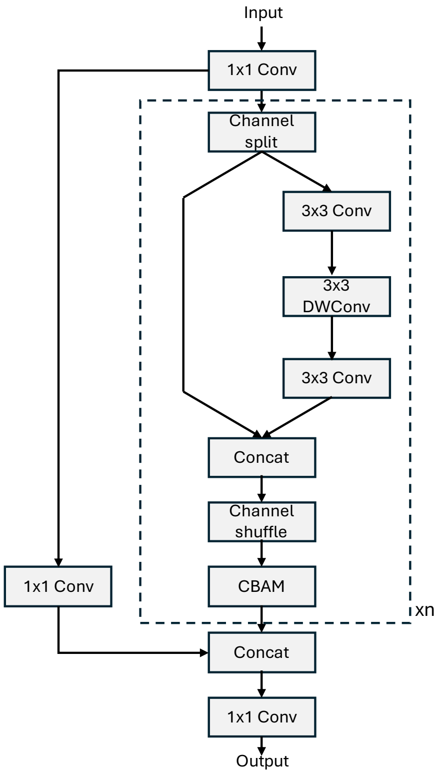

Fig 6. ShuffleNC3 structure

2.3. Improvement in Head

The Head is the part of object detection that mainly consists of convolutional layers, pooling layers, fully connected layers, etc. It is used to perform multi-scale object detection for large, medium, and small targets based on the features extracted from the Backbone part. The Anchor part is a pre-defined set of bounding boxes used to create candidate boxes on the feature map. Since the targets in the steel cable dataset used in this study are mostly medium and small-sized objects, only small target detection boxes [10, 13, 16, 30, 33, 23] and medium target detection boxes [30, 61, 62, 45, 59, 119] are defined. Experimental results have shown that using predefined anchor boxes yields better detection performance than using anchor boxes obtained through K-means clustering. Therefore, the network in this paper uses predefined anchor boxes instead of anchor boxes obtained through K-means clustering. The large target detection part and the corresponding Neck network part connected to the Backbone are pruned in the Head network, significantly reducing the computational complexity and parameter count of the network. The improved YOLOv5 network architecture is shown in Fig 7.

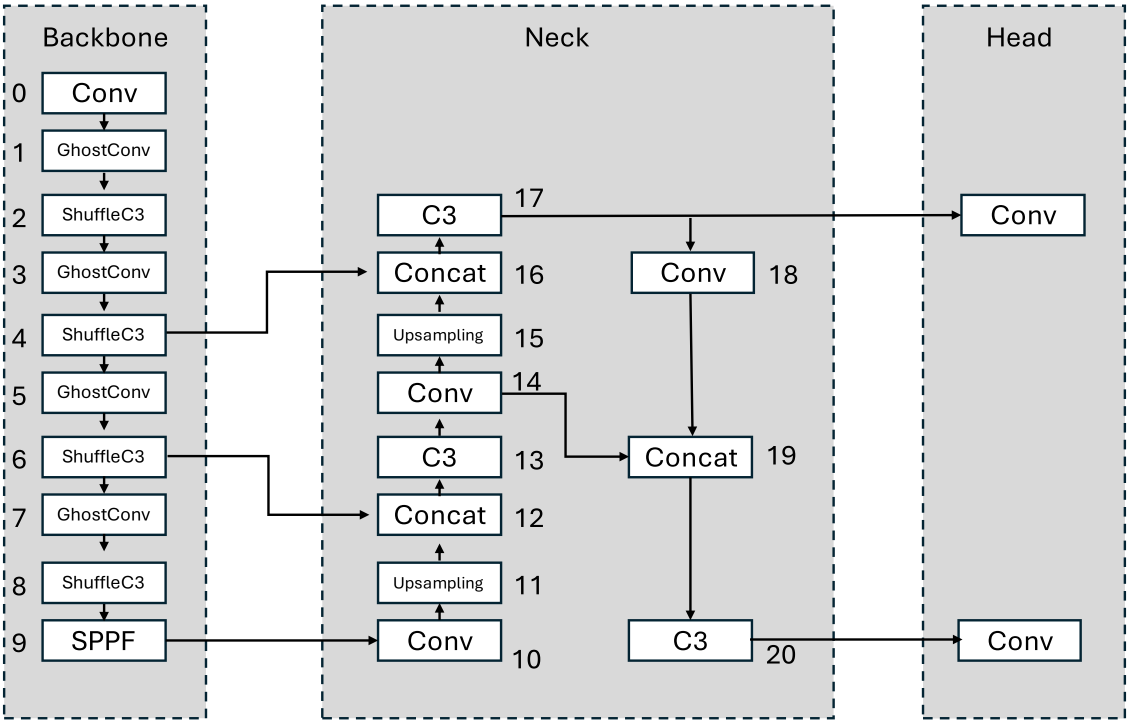

Fig 7. Improved Network Architecture

3. Experimental findings and subsequent analysis

In this section, the dataset, experimental settings, and evaluation metrics have been introduced and the improved network has been applied to the available datasets with experimental results and analysis. Subsequently, ablation experiments are conducted to verify the effects of the GhostConv module, ShuffleNC3 module, and Head part improvements. Finally, the proposed network method is compared with other commonly used defect recognition methods.

3.1. Dataset

To prove the effectiveness of the proposed improved method, we utilized the Cable Damage Computer Vision Project dataset and the DBTT Computer Vision Project dataset[15][16]. These datasets were partially integrated and subjected to data augmentation to create the Cable Damage dataset. The dataset includes two types of damage: fractures and burns in steel cables, consisting of a total of 6,590 images. The dataset was split into training (70%), validation (20%), and test sets (10%). Exactly, the training set contains 4,614 samples, while the validation set contains 1,314 samples. The training set plays a crucial role in training the network parameters to achieve the minimum loss function. The validation set is used to evaluate the accuracy of the trained network in recognizing surface damage. The two types of damage in the dataset are illustrated in Fig 8.

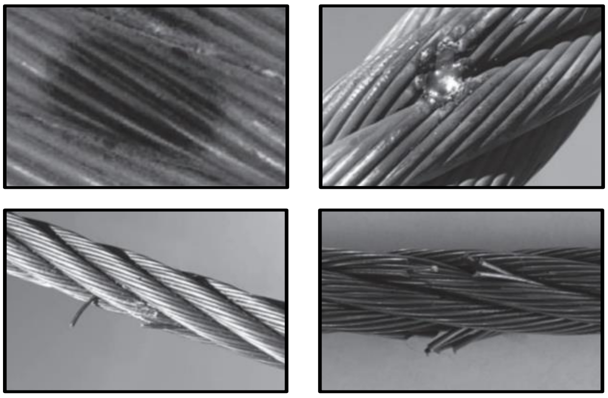

Fig 8. Burn and Fracture type of damage in the dataset

3.2. Experimental configuration

The experiments were conducted in the PyTorch framework. The hardware setup included an NVIDIA GeForce RTX 3080 GPU with 10 GB VRAM and an Intel(R) Xeon(R) 2.50 GHz Platinum 8255C CPU. The network was trained for 210 iterations, with a batch size of 32. Input images were resized to 640×640 and normalized. Stochastic Gradient Descent (SGD) was used as the optimizer with a linear learning rate schedule, starting at 0.01 and ending at 0.0001. The momentum parameter and weight decay coefficient were set to 0.937 and 0.0005, respectively. Since the network structure was modified, pre-training weights were not used in any of the experiments. The test results of the improved network are shown in Fig 9, indicating that the enhanced network accurately identifies surface damage in steel cables.

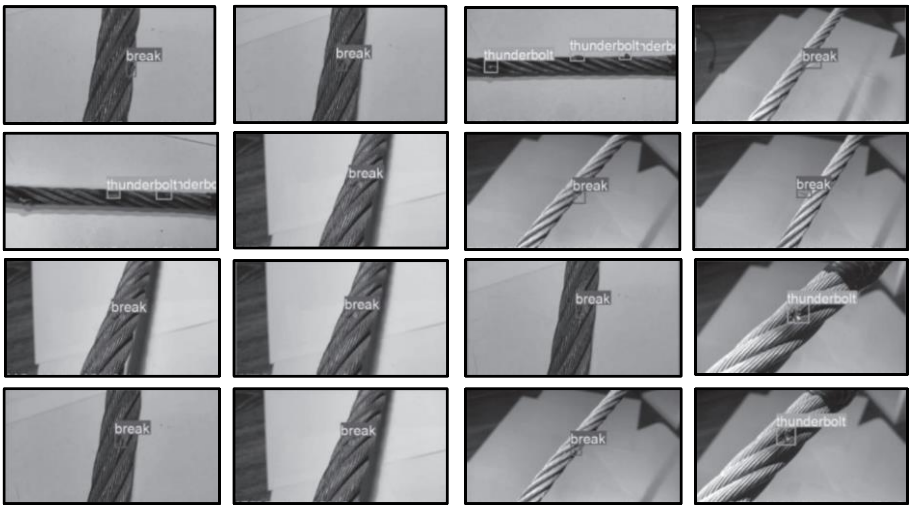

Fig 9. Test Results

3.3. Evaluation Indicators

This study evaluates the proposed network using metrics such as mean Average Precision (mAP), Precision, Recall, Floating-Point Operations per Second (GFLOPs), Parameters (Params), model size, and Frames Per Second (FPS) to provide a comprehensive assessment. In object detection tasks, Precision and Recall are crucial indicators to judge the recognition performance of the network, calculated as follows, where TP represents true positives, FP represents false positives, and FN represents false negatives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

The mean Average Precision (mAP) is the mean of average precision across all classes. Additionally, to compare the computational complexity of different networks, this study employs Floating-Point Operations (GFLOPs) and Parameters (Params) to represent performance differences between networks. Moreover, Frames Per Second (FPS) is used to indicate the inference speed, with the results being the average for 663 test images.

3.4. Control experiment results and analysis

This section of the experiment includes ablation studies to validate the effects of the GhostConv module, ShuffleNC3 module, and the introduction of attention mechanisms within ShuffleNC3. Four commonly used attention mechanisms, including SE-Net, ECA-Net, CA-Net, and CBAM-Net, were employed, and compared for their impact on the network. Furthermore, the study compares the proposed improved network with other state-of-the-art lightweight models, such as YOLOv3-Tiny, YOLOv6, YOLOv7-Tiny, and the two-stage object detection network Faster-RCNN. Additionally, popular lightweight backbone networks, MobileNetV3, ShuffleNetV2, and GhostNet, replace the default backbone of YOLOv5 for further comparison.

The results demonstrate that the introduced GhostConv module, ShuffleNC3 module, and improvements in the Head section significantly decrease the computational complexity and parameters of the network. The best fusion method in Experiment 7 shows a 1.1% increase in mAP, with a 43.4% reduction in parameters, a 31% reduction in computation, and a 42.3% reduction in model size. In attention mechanism comparison experiments, CBAM demonstrates slightly better execution in terms of mAP, and it is selected for integration into the ShuffleNC3 module.

In this section, the advantages of the GhostConv module, ShuffleNC3 module, and Head improvement in steel cable damage recognition are verified through ablation experiments. The experimental results are shown in Table 1 and graphical representation shown in Fig 10.

Table 1. Experimental results after various improvements

Experiment	GhostConv	ShuffleNC3	Head	mAP (%)	Precision (%)	Recall (%)	Params (M)	GFLOPs	Model Size (MB)	FPS
1	—	—	—	83.6	87.5	80.3	7.02	15.8	14.4	106.3
2	√	—	—	83.7	88.3	80.2	6.24	14.0	12.8	112.3
3	—	√	—	84	90.9	78.6	6.52	14.1	13.5	61
4	—	—	√	83.9	91.3	78.2	5.23	14.3	10.8	137
5	√	√	—	84.1	88.7	81.6	5.75	12.3	11.9	57.8
6	√	—	√	84.3	88.9	80.4	4.46	12.5	9.2	126.6
7	√	√	√	84.7	88.9	82.1	3.97	10.9	8.3	64.5

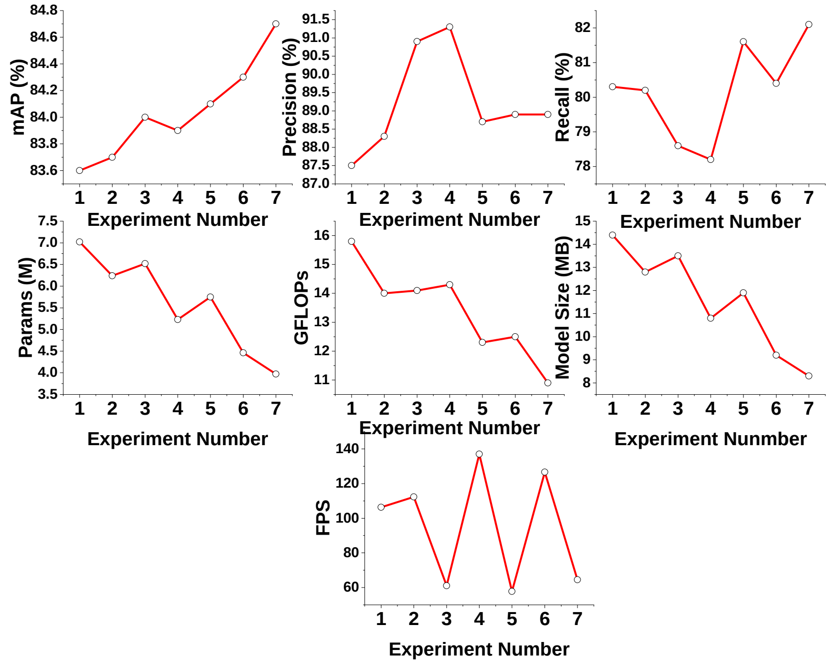

Fig 10. Experimental results after various improvements

The ShuffleNC3 module introduces an attention mechanism. Currently, the most commonly used attention mechanisms include SE-Net, ECA-Net, CA-Net[13], and CBAM-Net, a total of four types. These four attention mechanisms are individually incorporated for comparative experiments, as shown in Table 2 and graphical representation shown in Fig 11.

Table 2. Results of each attention mechanism

Experiment	SE	ECA	CA	CBAM	mAP%	Precision (%)	Recall (%)	Prams (M)	GFLOPs	Model size (MB)
1	√	—	—	—	83.9	87.8	82.1	3.97	10.9	8.3
2	—	√	—	—	83.5	88.5	82.7	3.94	10.9	8.2
3	—	—	√	—	84.1	88.1	81.9	3.97	10.9	8.4
4	—	—	—	√	84.7	88.9	82.1	3.97	10.9	8.3

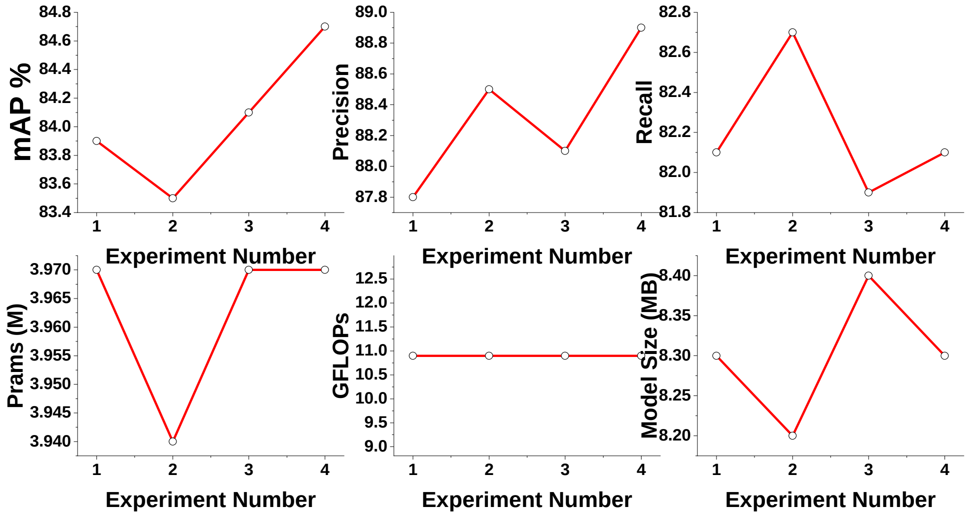

Fig 11. Recognition results of each attention mechanism

To justify the performance of the improved network in steel cable damage recognition, we compared it with other advanced lightweight network models, including YOLOv3-Tiny, YOLOv6[18], YOLOv7-Tiny. Additionally, performance comparisons were conducted with commonly used two-stage object detection networks like Faster-RCNN[19]. Furthermore, commonly used lightweight backbone networks such as MobileNetV3, ShuffleNetV2, and GhostNet[20], [21] were employed to replace the default Backbone of YOLOv5 for comparison. The results are presented in Table 3 and graphical representation shown in Fig 12.

Table 3. Performance comparison of commonly used target detection networks

Experiment	Model	mAP (%)	Params (M)	GFLOPs	Model size (MB)
1	YOLOv5s	83.6	7.02	15.8	14.4
2	YOLOv3-Tiny	84.1	8.67	13	16.6
3	YOLOv6s	81.9	18.5	45.3	181
4	YOLOv7-Tiny	83.4	6.01	13.2	12.3
5	Faster-RCNN	61.3	137.1	370.2	108
6	MobileNetV3-YOLOv5	82.5	5.02	11.3	10.4
7	ShuffleNetV2-YOLOv5	81.7	3.79	8.0	8.0
8	GhostNet-YOLOv5	84.2	4.76	7.9	10.0
9	Improved YOLOv5	84.7	3.97	10.9	8.3

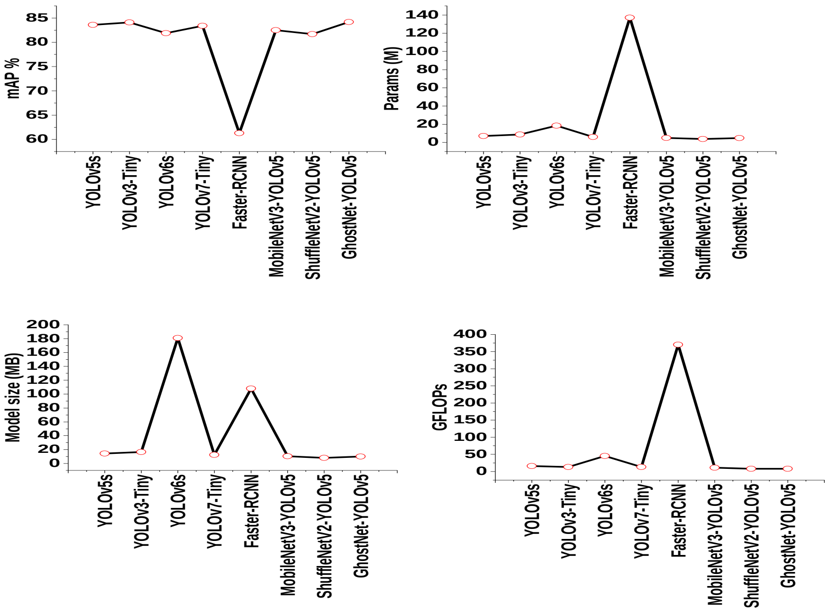

Fig. 12 Performance comparison of commonly used target detection networks

Based on ablation experiments, after introducing GhostConv module separately, ShuffleNC3 module, and Head improvement separately resulted in a significant reduction in computational and parameter quantities. Experiment number 5 and 7 involved different methods of combining above three improvements. Experiment number 7 shows that fusion method exhibited the best performance, with a 1.1% increase in average accuracy. Parameter and computational quantities were reduced by 43.4% and 31%, respectively, and the model size decreased by 42.3%. Since the ShuffleNC3 module incorporates the CBAM attention mechanism, it may have some impact on the network's computational speed. The FPS metric is influenced by experimental configurations, yielding different results on various devices. Therefore, this paper places more emphasis on average accuracy and network computational costs, making Experiment number 7 network model more suitable.

Through attention comparison experiments, it is observed that the four attention mechanisms have a similar impact on parameter quantity, computational cost, and model size. Using CBAM can achieve higher average accuracy. Therefore, CBAM is incorporated into the ShuffleNC3 module in this network. In the comparison experiments with different models, although YOLOv3-Tiny has slightly higher average accuracy than YOLOv5s, its use of DarkNet-53 as the backbone results in higher computational costs. On the other side YOLOv6s and YOLOv7-Tiny perform worse in steel cable damage recognition compared to the pre-improved YOLOv5s. The proposed network maintains high average accuracy while outperforming other models in computational costs and model size. Among the models with replaced backbones, ShuffleNetV2-v5 and GhostNet-v5 networks, due to their extensive use of lightweight convolutions and shallower network depth, exhibit slightly lower parameter and computational quantities than the improved network. ShuffleNetV2-v5 reduces parameters by 0.18M and computations by 2.9G, while GhostNet-v5 reduces computations by 3G. However, their shallower networks cannot guarantee the accuracy of recognition, resulting in lower average accuracy compared to the improved network. Therefore, the improved network is more in line with the requirements of this paper, exhibiting the best performance.

4. Conclusion

In this paper, a lightweight network dedicated to the recognition of surface damage on steel cables installed in public building, based on YOLOv5, has been proposed. Based on the proposed network, it reduced the network parameters and computational complexity, saved computational cost, and ensure the average detection accuracy, GhostConv modules, ShuffleNC3 modules and improvement in Head section were employed. Tested on the Cable Damage dataset, the improved network achieves a 1.149% increase in average accuracy compared to YOLOv5s, with reductions of 43% in parameters and 31% in computation complexity, respectively, and the model size also decreased by approximately 42%.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Zhenhua Yu and Dr. Zhanli Li, their expertise, encouragement, and commitment to excellence have played a pivotal role in shaping the direction of this conference paper. I am also thankful to my Computer Science Department, Xian University of Science and Technology, Xian, China, which has created an environment conducive to research and learning. Additionally, I extend my appreciation to my colleagues and peers for their constructive discussions and collaborative efforts.

Author contributions

Muhammad Wahab Hanif: Visualization and Experiments, Data Validation, Software, Methodology. Preparation of Original Draft. Conceptualization, Methodology, Software, Field study **Dr.Zhanli Li:** Conceptualization, Data Correction and curation, Review of Draft. Field study. **Dr. Rehmat Bashir:** Visualization, Investigation, Data Preparation, Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.

[2] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles." arXiv, Jan. 03, 2023. doi: 10.48550/arXiv.2112.11798.

[3] "Algorithms | Free Full-Text | A Comprehensive Analysis of Real-Time Car Safety Belt Detection Using the YOLOv7 Algorithm." Accessed: Jan. 12, 2024. [Online]. Available: https://www.mdpi.com/1999-4893/16/9/400

[4] N. Mohod, P. Agrawal, and V. Madaan, "YOLOv4 Vs YOLOv5: Object Detection on Surveillance Videos," in *Advanced Network Technologies and Intelligent Computing*, I. Woungang, S. K. Dhurandher, K. K. Pattanaik, A. Verma, and P. Verma, Eds., in Communications in Computer and Information Science. Cham: Springer Nature Switzerland, 2023, pp. 654–665. doi: 10.1007/978-3-031-28183-9_46.

[5] F. Ni and Y. Yao, "Multi-Task Learning via Scale Aware Feature Pyramid Networks and Effective Joint Head," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0. Accessed: Jan. 12, 2024. [Online]. Available: https://openaccess.thecvf.com/content_ICCVW_2019/html/AUTONUE/Ni_Multi-Task_Learning_via_Scale_Aware_Feature_Pyramid_Networks_and_Effective_ICCVW_2019_paper.html

[6] Y. Jiang, Z. Tan, J. Wang, X. Sun, M. Lin, and H. Li, "GiraffeDet: A Heavy-Neck Paradigm for Object Detection," presented at the International Conference on Learning Representations, Oct. 2021. Accessed: Jan. 12, 2024. [Online]. Available: https://openreview.net/forum?id=cBu4ElJfneV

[7] Z. Gao, G. Yang, E. Li, and Z. Liang, "Novel Feature Fusion Module-Based Detector for Small Insulator Defect Detection," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16807–16814, Aug. 2021, doi: 10.1109/JSEN.2021.3073422.

[8] M. Qiu, L. Huang, and B.-H. Tang, "ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion," *Remote Sensing*, vol. 14, no. 14, Art. no. 14, Jan. 2022, doi: 10.3390/rs14143498.

[9] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019, doi: 10.1109/ACCESS.2019.2904620.

[10] S. Zhang, S. Che, Z. Liu, and X. Zhang, "A real-time and lightweight traffic sign detection method based on ghost-YOLO," *Multimed Tools Appl*, vol. 82, no. 17, pp. 26063–26087, Jul. 2023, doi: 10.1007/s11042-023-14342-z.

[11] G. Yaqing and B. Ge, "Research on Lightweight Convolutional Neural Network in Garbage Classification," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 781, no. 3, p. 032011, May 2021, doi: 10.1088/1755-1315/781/3/032011.

[12] Z. Fan, W. Hu, H. Guo, F. Liu, and D. Xu, "Hardware and Algorithm Co-Optimization for pointwise convolution and channel shuffle in ShuffleNet V2," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2021, pp. 3212–3217. doi: 10.1109/SMC52423.2021.9659057.

[13] "LAM: Lightweight Attention Module | SpringerLink." Accessed: Jan. 12, 2024. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-10986-7_39

[14] J. Wang, J. Yu, and Z. He, "DECA: a novel multi-scale efficient channel attention module for object detection in real-life fire images," *Appl Intell*, vol. 52, no. 2, pp. 1362–1375, Jan. 2022, doi: 10.1007/s10489-021-02496-y.

[15] X. Ji and Y. Niu, "A Lightweight Network for Human Pose Estimation Based on ECA Attention Mechanism," *Electronics*, vol. 13, no. 1, Art. no. 1, Jan. 2024, doi: 10.3390/electronics13010150.

[16] "DBTT Dataset > Overview," Roboflow. Accessed: Jan. 14, 2024. [Online]. Available: https://universe.roboflow.com/cv-proj/dbtt

[17] "cable damage Dataset > Overview," Roboflow. Accessed: Jan. 14, 2024. [Online]. Available: https://universe.roboflow.com/roboflow-100/cable-damage

[18] T. Song *et al.*, "Lightweight detection network based on receptive-field feature enhancement convolution and three dimensions attention for images captured by UAVs," *Image and Vision Computing*, vol. 140, p. 104855, Dec. 2023, doi: 10.1016/j.imavis.2023.104855.

[19] "YOLO-ERF: Lightweight Object Detector for UAV Aerial Images." Accessed: Jan. 15, 2024. [Online]. Available: https://www.researchsquare.com

[20] J. Huang, J. Chen, and H. Wang, "A lightweight and efficient one-stage detection framework," *Computers and Electrical Engineering*, vol. 105, p. 108520, Jan. 2023, doi: 10.1016/j.compeleceng.2022.108520.

[21] J. Yuan, N. Wang, S. Cai, C. Jiang, and X. Li, "A Lightweight Crack Detection Model Based on Multibranch Ghost Module in Complex Scenes," *IEEE Sensors Journal*, vol. 23, no. 19, pp. 22754–22762, Oct. 2023, doi: 10.1109/JSEN.2023.3300715.