Research on Improving YOLOv5s Algorithm for Defect Detection in Cylindrical Coated Lithium-ion Batteries

Ziqiang Lin, Lijun Zhu, Jinyu Zhang, Yuanhang Zhang, Xudong Liu

***Abstract*—The advancement of new energy vehicles has led to more demanding standards for detecting defects in cylindrical coated lithium batteries. The current research lacks robustness and has low performance. This paper seeks to provide real-time defect identification in cylindrical coated lithium batteries and improve the object detection method of the YOLOv5s model. This paper presents an MGSEC3 module with multi-scale feature extraction and integration of the SENet network in the YOLOv5 Backbone network. This module aims to reduce computational burden and enhance feature extraction efficiency as much as possible. At the same time, the CARAFE operator is utilized to enhance the up-sampling operator in order to reduce the loss of feature information. In addition, enhancements to the loss function enhance both the detection performance and convergence speed. The enhanced YOLOv5s model achieved an average detection rate of 82.4% on the custom cylindrical coated lithium battery dataset, 2.3% higher than the original YOLOv5s. This paper significantly enhances the precision and efficiency of flaw identification in cylindrical coated lithium batteries.**

***Index Terms*—Lightweight model, Attention Mechanism, Defect detection, Image processing**

I. Introduction

CYLINDRICAL coated lithium batteries have been around for a long time and are used in many electronic products. The advancement of lithium batteries has been significantly boosted by the growth of the new energy automotive sector. The industry's rapid expansion makes it crucial to determine if cylindrical coated lithium batteries have any problems. The flaws might impact the effectiveness, dependability, and lifespan of the battery. It can lead to operational concerns and pose severe safety risks when utilized. Moreover, these flaws will directly impact how people perceive and assess electronic items like new energy cars. Thus, identifying defects in cylindrical coated lithium batteries is essential to maintaining product quality and safety and boosting user trust in electronic devices.

Artificial visual inspection is by far the most common method for discovering defects in cylindrical coated lithium batteries. However, this method has a high error rate and low efficiency and may cause pain to workers. Machine vision technology is being used to detect defects in many items. There are two main research paths for machine vision algorithms: defect recognition using traditional machine learning and defect detection using deep learning. X. R. Zeng[1] used the Sobel template algorithm to detect the surface of cylindrical bare-shell batteries. Z. C. Kang[2] suggested a pit detection approach that combines density detection, gaussian gradient convolution, and the gray feature. Y. Li[3] employed median filtering and the 3σ criterion to detect membrane rupture in cylindrical coated lithium batteries. C. X. Liu[4] suggested a least squares method for pit detection in cylindrical coated lithium batteries. S. T. Guo[5] initially derived the gray difference curve of the battery's outer surface and subsequently identified the inflection point on the gray distribution curve by a tailored gray difference model, enabling the detection of pits. Traditional machine learning defect detection methods effectively identify defects with consistent and singular characteristics. However, due to their subtle nature and distinct shapes, they struggle to detect surface defects on cylindrical coated lithium batteries, resulting in poor detection reliability.

The other uses deep learning techniques to detect defects. X. Y. Feng[6] conducted a comparative investigation on the detection of surface flaws in battery steel shells using three models: Fast R-CNN, Cascade R-CNN, and YOLOv3. J. Tian[7] et al. suggested a lightweight YOLOv4 network model for identifying flaws in the lithium battery steel shell's surface. First, the prior frames in the data set are re-clustered using the k-means++ algorithm. Second, MobileNetv1 is substituted for the CSPDarknet53 module to enhance the YOLOv4 model. H. B. Xu[8] performed studies with the YOLOv3 model. The K-means technique optimizes the data set in the preceding box. Then, they used ablation to reduce the complexity of the network. Finally, The YOLOv3-x model is proposed to detect lithium battery defects. Y. Q. Gui[9] et al. conducted defect detection of lithium batteries based on the YOLOv4 model. First, the conventional convolution in the CSPDarknet53 module has been substituted with the dilated convolution. Second, the channel

Manuscript received January 31, 2024; revised Jun 11, 2024.

Ziqiang Lin is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China. (e-mail: 591913784@qq.com).

Lijun Zhu is a Professor of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China. (Corresponding author to provide phone: +86-159-9818-3125; e-mail: zhulijun@yeah.net).

Jinyu Zhang is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China. (e-mail: 2899725854@qq.com).

Yuanhang Zhang is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China. (e-mail: 805447267@qq.com).

Xudong Liu is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, China. (e-mail: 114cris@gmail.com).

attention mechanism is included in the Neck network. Meanwhile, the CondConv is integrated into classification and boundary box regression loss functions. The mentioned models exhibit deficiencies, including model redundancy and sluggish detection speed.

II. RELATED WORK

Deep learning-based object detection algorithms are now classified into two groups: one-stage and two-stage object detection methods. The one-stage object detection technique performs object detection in a single stage by transforming the object detection task into a regression problem. This is accomplished by anticipating both the object's bounding box and category information at specific places or anchor points inside the image. The one-stage object detection technique provides faster speed and lower computing costs due to its direct processing of the input image. It is ideal for real-time applications that handle extensive datasets. Representative one-stage object detection algorithms include the YOLO (You Only Look Once) series[10-14] and the SSD (Single Shot MultiBox Detector) algorithm[15].

The two-stage object detection technique breaks down the task into two steps for processing. Initially, it creates a group of potential regions or identifies regions of interest (ROI) on the feature map. The ROI may include object projects. Then, the object detection results are established by classification and boundary box regression. This method is better suited for detecting complicated sceneries and small objects than the one-stage algorithm. Representative algorithms include the R-CNN series[16-18].

YOLOv5 is an improved one-stage object detection algorithm that Ultralytics developed based on the YOLOv4 network structure. Released in June 2020, it has now advanced to version 7.0. YOLOv5 features a more streamlined network structure than YOLOv4, leading to lighter files, improved detection accuracy, and faster processing performance. YOLOv5 provides four distinct network architectures: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The parameters and network size go from YOLOv5s to YOLOv5x. The choice of structure depends on specific application scenarios and resource constraints.

The YOLOv5s model comprises four primary network components: Input, Backbone, Neck, and Prediction. The Input network performs operations like Mosaic data augmentation, adaptive image scaling, and adaptive anchor frame calculation on input images. The Backbone network utilizes the CBS module and C3 module to extract features from the input image. Then, the SPPF module normalizes the feature scale and expands the feature's receptive field. The Neck network integrates the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN) modules to accomplish multi-scale feature fusion. This enables feature maps of varying scales to encompass abundant semantic and location information. Finally, the Prediction network has detection heads of three scales to assess the final feature map provided by the feature fusion layer for prediction results.

III. ALGORITHM DESIGN

A. Efficient MGSEC3 Module

The C3 module functions as the key module of YOLOv5s in the Backbone network. It comprises two parallel branches: the first branch, which is through a CBS module and then stacks several residual network modules to extract high-level semantic information. Subsequently, the second branch also passes through a CBS module before being connected to the output of the first branch. The combined output is subsequently processed by another CBS module for feature fusion, resulting in the fused features. Fig.1 illustrates the structure of the CBS module, and Fig.2 depicts the C3 model structure.

Real-time detection is impossible because of the numerous parameters in the C3 module. We suggest a lightweight C3 module named MGSEC3, which utilizes the GhostBottleneck structure to tackle this problem. First, we add three convolution structures horizontally to the GhostBottleneck structure. This structure integrates DWConv and Conv layers to extract sophisticated semantic features, improving the model's feature extraction capacity. The GhostBottleneck structure is illustrated in Fig.3. To capture multi-scale features, we set the convolution kernels of the three DWConv modules as 3x3, 5x5, and 7x7 to extract parallel features at each scale. Finally, we add the output of these branches to form a multi-scale GhostBottleneck (MGhostBottlenack) structure. The MGhostBottleneck structure is shown in Fig.4.

In the Backbone network, when shortcut1 and shortcut2 are enabled at the same time, it is named the MGhostBottleneck_1 module. In contrast, only the shortcut1 branch exists in Neck network architectures referred to as MGhostBottleneck_2. We have improved the handling of multi-scale features by incorporating the MGhostBottleneck structure during feature extraction and fusion, leading to a more robust high-level semantic grasp of the tasks. The MGC3 module substitutes the bottleneck architecture of the prior C3 module with the MGhostBottleneck.

The SENet[19] attention module was incorporated after the final CBS convolution of the MGC3 module to boost the model's focus on channel connections. The SENet incorporates a distinctive attention mechanism that autonomously learns and assesses the significance of individual feature channels. The importance weights are utilized to improve the representation of significant

Fig. 1 Structure of the CBS

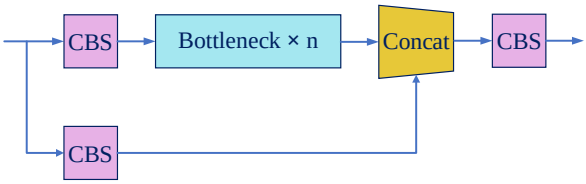

Fig. 2 Structure of the C3

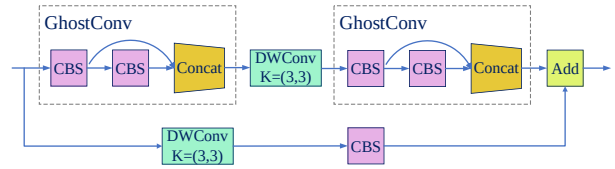

Fig. 3. Structure of the GhostBottleneck

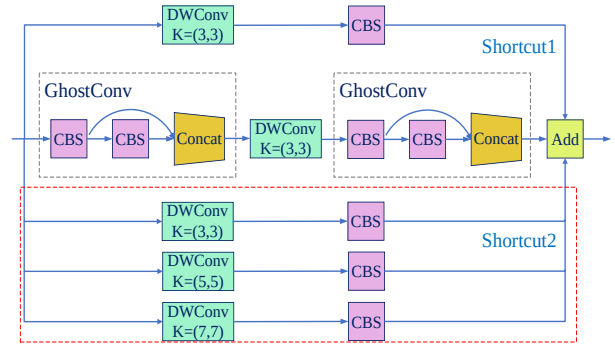

Fig. 4. Structure of the MGhostBottleneck

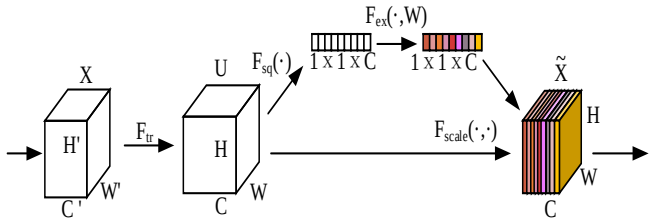

Fig. 5. Structure of the SENet

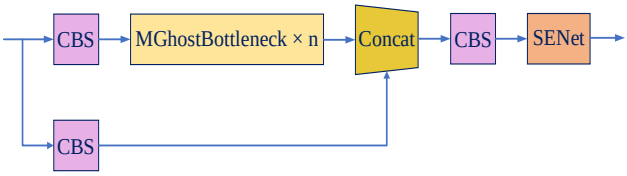

Fig. 6. Structure of the MGSEC3

information. Meanwhile, it enhances the perception of essential features by dynamically modifying channel weights and boosting the model's performance. The structure of SENet is illustrated in Fig.5.

Integrating SENet into the MGC3 module results in the creation of the MGSEC3 module, enhancing the model's learning capacity. The MGSEC3 module structure is shown in Fig.6.

B. CARAFE Upsampling Operator

The Neck network in YOLOv5s utilizes nearest neighbor interpolation for upsampling. This method involves duplicating the values of neighboring pixels while increasing the resolution, without completely taking into account the semantic details of feature mapping. Since this method may miss certain fundamental properties and ignore the link between pixels. Thus, it can result in a decrease in the quality of upsampling feature mapping, particularly when detecting small objects or complicated scenarios.

In order to tackle this problem, the CARAFE[19] lightweight generic upsampling operator is presented to replace the nearest neighbor interpolation method. The CARAFE program enhances the production of superior upsampled feature maps. The CARAFE consists of two main modules: the upsampling kernel prediction module and the feature reorganization module. Two modules collaborate to enhance the upsampling process, hence boosting the model's performance. The CARAFE operator structure is shown in Fig.7.

The upsampling kernel prediction module analyzes the input feature map to predict the necessary upsampling kernel for feature points at various positions. It includes channel compression, content-encoding, and kernel normalization submodules to generate adaptive upsampling kernels for different upsampling rates. The content encoding submodule encodes the feature map using multiple convolutional layers to provide an upsampling kernel with a shape of $H \times W \times \sigma^2 \times k_{up}^2$. Where σ is the upsampling multiplier and k_{up} is the size of the upsampling kernel for a single feature point. The upsampling kernel prediction module takes into account the deep semantic information of the input feature map. This operation creates upsampling kernels that can adjust to various circumstances.

The feature reorganization module plays a vital role by mapping the model's output feature points to the input feature map. It combines these points with the upsampling kernel and local area information using dot product operations to create more semantically informative upsampling feature maps. This improves the quality of feature maps and enables the model to capture and utilize important information in the image effectively.

Compared with the nearest neighbor interpolation method, the CARAFE upsampling operator enhances semantic information integration. Meanwhile, it improves the model's perception and understanding of image details and critical features. The CARAFE can adapt to various feature points' upsampling requirements and minimize information loss. The model can obtain improved quality and accuracy in object identification tasks by implementing the CARAFE upsampling operator.

C. SIoU Loss function

The YOLOv5s model utilizes CIoU Loss in the loss function. Which takes into account the width-to-height ratio of the regression box and the distance between the centers of the real box and the predicted box. However, it does not consider the direction information between the real box and the predicted box. This leads to a slower convergence speed. SIoU Loss[21] is introduced to account for the vector angle between the real box and the forecast box in order to tackle this problem. The SIoU Loss comprises four components: angle loss, distance loss, shape loss, and IoU loss.

Utilizing SIoU Loss for bounding box regression enhances the model's ability to account for angle information between the actual box and the predicted box. This leads to faster convergence and ultimately results in improved performance and more precise detection outcomes in object identification.

The angle loss is shown in Fig.8(a):

It is defined as:

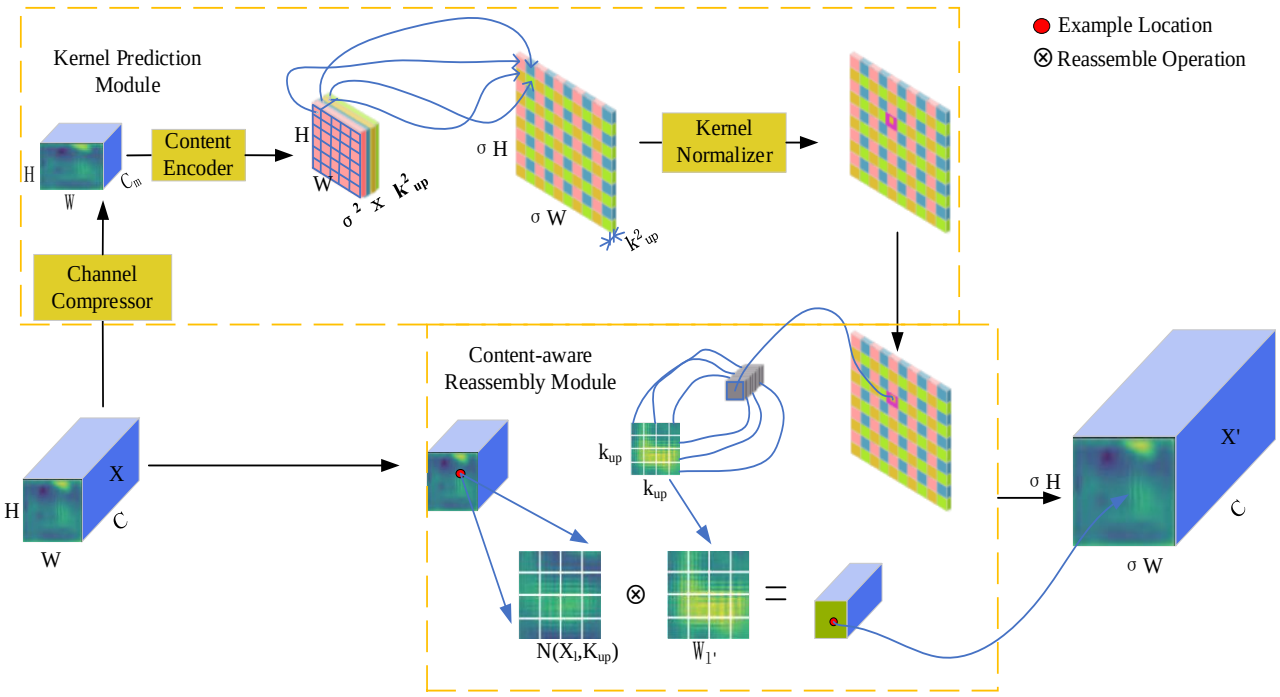

Fig. 7. Structure of the CAFARE

$$\Lambda = 1 - 2 * \sin^2(\arcsin(\frac{Ch}{\sigma}) - \frac{\pi}{4}) \quad (1)$$

In the definition, Ch is the height difference between the center points of the real box and the predicted box, and σ is the distance between the center points of the real box and the predicted box.

The distance loss is shown in Fig.8(b):

It is defined as:

$$\Delta = 2 - e^{-\gamma\rho x} - e^{-\gamma\rho y} \quad (2)$$

In the definition:

$$\rho x = (\frac{b_{cx}^{gt} - b_{cx}}{c_w})^2, \rho y = (\frac{b_{cy}^{gt} - b_{cy}}{c_h})^2, \gamma = 2 - \Lambda \quad (3)$$

C_w , C_h are the width and height of the minimum bounding rectangle between the true and predicted boxes. b_{cx}^{gt} and b_{cy}^{gt} are the true box center coordinates, while b_{cx} and b_{cy} are the predicted box center coordinates.

Shape loss is defined as:

$$\Omega = \sum_{t=w,h} (1 - e^{-w_t})^\theta = (1 - e^{-w_w})^\theta + (1 - e^{-w_h})^\theta \quad (4)$$

In the definition:

$$w_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})}, w_y = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \quad (5)$$

w, h, w_{gt}, h_{gt} are the width and height of the predicted box and the ground truth box, respectively, controlling the degree of attention paid to shape loss.

The IoU loss is shown in Fig.9. Among them, set A is the intersection of the real box and the predicted box, and set B is the union of the real box and the predicted box.

To sum up, the final definition of the SIoU Loss loss function is:

$$L_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \quad (6)$$

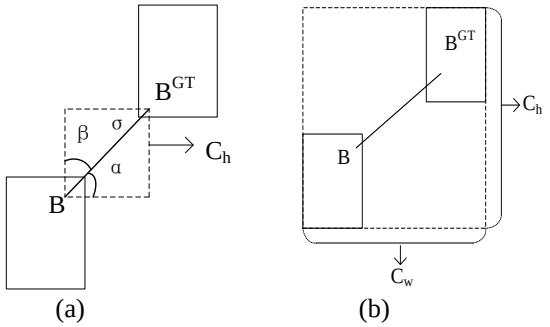

Fig. 8. (a) Angle cost; (b) Distance cost

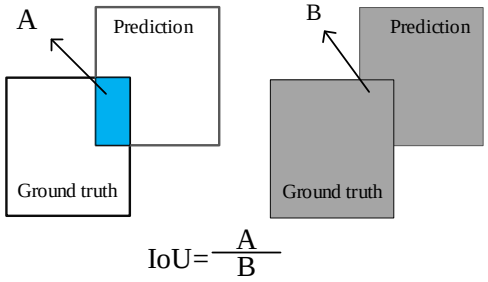

Fig. 9. IoU cost

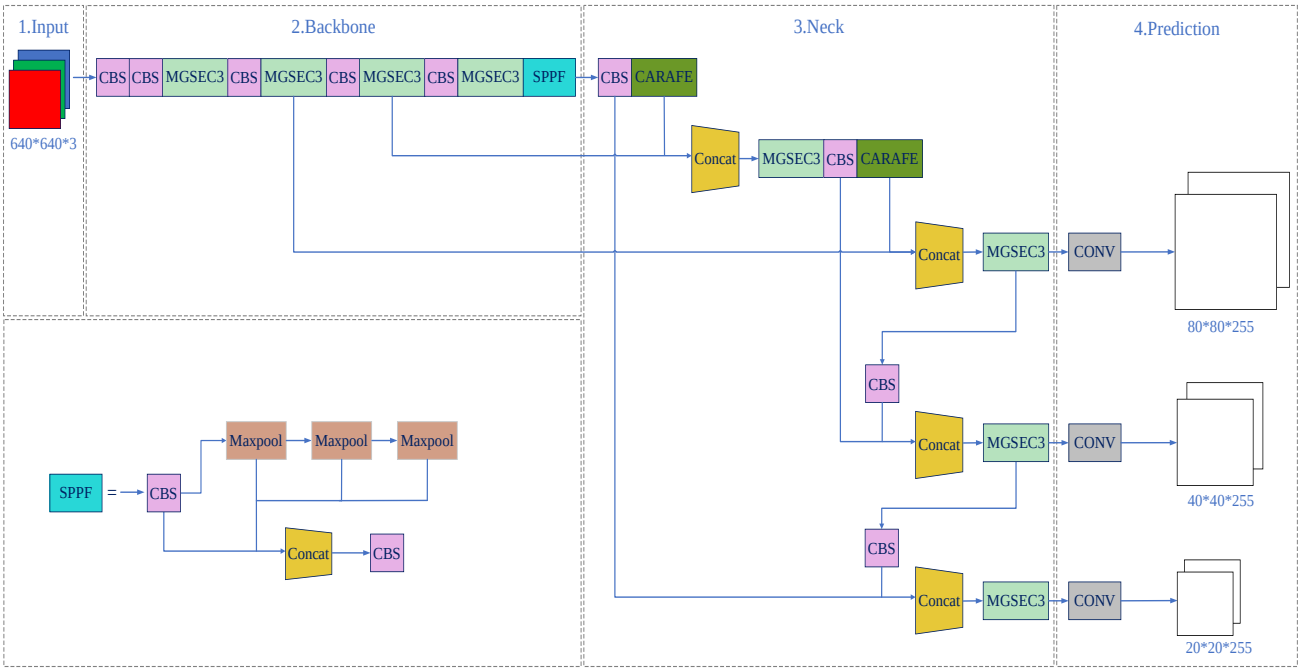

Fig. 10. MCS-YOLO Network Model Structure

TABLE I
NUMBER OF VARIOUS DEFECTS

Defects	Pit	Rupture
Number	796	3854

D. Overall Architecture of MCS-YOLO Model

We suggest enhancing the MGSEC3 module of the Backbone network. Meanwhile, we propose improving the upsampling operator of the Neck network and the loss function in the original YOLOv5s model. The MCS-YOLO model structure is shown in Fig.10.

IV. EXPERIMENT

A. Experiment Dataset

We study defects such as pits and ruptures on the cylindrical surface of thin film-coated lithium batteries. We captured images in a real field environment, taking six photos at 60° intervals for each battery to ensure a clear and complete depiction of circumferential surface defects. A total of 7024 pictures were collected, each with dimensions of 940 x 330. The images were then filled with background and resized to 640 x 640. Subsequently, we used the Labelme annotation tool to label the collected images, converting the comments into YOLOv5-compatible TXT format. Out of the 3458 labeled pictures, Table 1 shows the number of pits and broken film defects. The images were split into a training set and a test set with a ratio of 8:2.

B. Experimental Environment and Parameter Setting

All experiments were conducted in a computing environment running on the Ubuntu 18.04 operating system. The system has 16GB of memory, an Intel Core i7-9700K processor, 8 CPU cores, and an NVIDIA GeForce RTX 2070 graphics card with a memory capacity of 8GB. PyTorch version 1.13.0 was used in the experiment, alongside CUDA version 11.3 for GPU acceleration.

The input image size of the algorithm is 640 × 640, and the Warmup method is used to warm up the model. The cosine learning rate decay strategy is used to train the network. The initial learning rate is 0.01, and the last round of the learning rate decay ratio is 0.01. The training batch size is 16, and the total number of epochs is 300. The stochastic gradient descent optimizer SGD is used to iterate the network parameters, with a weight decay rate of 0.0005 and a momentum factor of 0.937.

TABLE II
RESULT OF COMPARISON WITH THE YOLOV5S MODEL

Model	Size	m*AP*	Params	FLOPs
YOLOv5s	640×640	80.1	7.016	15.8
+MGSEC3	640×640	81.7(+1.6)	5.240	11.6
+CARAFE	640×640	80.3(+0.2)	5.380	11.9
+SIoU	640×640	80.7(+0.6)	5.257	11.7
MCS-YOLO	640×640	**82.4**(+2.3)	5.347	12.0

TABLE III
RESULT OF COMPARISON WITH THE OTHER YOLOV5S LIGHTWEIGHT MODELS

Model	Size	m*AP*	Params	FLOPs
YOLOv5s	640×640	80.1	7.016	15.8
YOLOv5s-MobileNetv3	640×640	78.2	3.604	6.4
YOLOv5s-ShuffleNetv2	640×640	79.6	5.620	11.6
YOLOv5s-Ghostnet	640×640	80.0	5.257	11.7
YOLOv5s-MGSEC3	640×640	**81.7**	5.347	12.0

TABLE IV
RESULT OF COMPARISON WITH THE OTHER ATTENTION MECHANISM NETWORK

Model	Size	m*AP*	Params	FLOPs
YOLOv5s	640×640	80.1	7.016	15.8
YOLOv5s+MGC3+CBAM	640×640	81.2	5.160	11.2
YOLOv5s+MGC3+ECA	640×640	80.8	5.330	11.6
YOLOv5s+MGC3+CA	640×640	80.6	5.377	11.4
YOLOv5s+MGC3+SimAM	640×640	81.0	5.337	11.9
YOLOv5s-MGSEC3	640×640	**81.7**	5.347	12.0

C. Evaluation Measures

In order to comprehensively and objectively evaluate the performance of the training model, we select three evaluation indexes: mAP, Params, and GFLOP.

mAP is the mean average precision. The mAP is a metric used to measure the accuracy of a model, which takes into account each category of data and calculates its average accuracy. Generally speaking, the higher the accuracy of the model, the better the model performs on the task. The formula for calculating mAP is as follows:

$$mAP = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} AP_i = \frac{1}{n} \sum_{i=1}^{n} \int_0^1 P(R) dR \tag{7}$$

In this statement, n signifies the number of categories, and P represents precision. P is a metric that quantifies the accuracy of a model by comparing the number of correctly predicted positive samples to the total number of anticipated positive samples. Simply put, it indicates the model's precision in recognizing positive instances. How many true positive instances does it accurately recognize? The model's performance is evaluated by calculating the mean average precision (mAP) of overall defect detection. Which is obtained by averaging the recognition accuracy of each category under various intersection-over-union (IoU) criteria. Evaluating the mAP index allows for a more thorough assessment of the model's performance, serving as a crucial guide for model selection and optimization.

D. Visualization Analysis

Fig.11 displays the detection performance comparison between the YOLOv5s model and the MCS-YOLO model. The experimental results indicate that object detection utilizing YOLOv5s in Fig.11(a)(b)(c) shows a tendency to overlook items. The MCS-YOLO method recognizes all items in the scene as shown in Fig.11(d), (e), (f). Meanwhile, it has shown a significant enhancement in detection accuracy when compared to the YOLOv5s algorithm. In conclusion, our model demonstrates clear superiority in detecting faults on the circular surface of cylindrical coated lithium batteries. The MCS-YOLO model can identify and capture faults more accurately than the YOLOv5s model. That also significantly enhances the efficiency and accuracy of defect identification.

E. Ablation Studies

The detection performance of each enhanced module in the MCS-YOLO algorithm was evaluated by comparing the MCS-YOLO model with the original YOLOv5s model. To confirm the advanced features of the proposed MGSEC3 module, it will be compared with other lightweight modules and attention mechanism structures. The MCS-YOLO is evaluated against other YOLOv5s lightweight models and different attention mechanism network models. Each detection algorithm operates in consistent experimental settings and utilizes identical training and test data sets.

1) Comparison with the YOLOv5s model

We performed ablation experiments on a dataset with a consistent environment and parameter settings to assess the performance of each approach in MCS-YOLO. Table II displays the outcomes of the ablation studies, with the best accuracy highlighted in bold. In Table II, MGSEC3 indicates that the MGSEC3 module replaces the C3 module; CARAFE represents the replacement of the upsampling operator from nearest neighbor interpolation upsampling to CARAFE operator; SIoU means replacing the boundary loss function with SIoU loss function.

As depicted in Table II, the introduction of the MGSEC3 module resulted in a 4.2Gc reduction in FLOP and a 1.6% increase in mAP for MCS-YOLO. The incorporation of MGSEC3 not only significantly decreased the computational

redundancy of the model. But it also effectively improved the model's understanding of object context, thereby enhancing its recognition accuracy. It can effectively facilitate the extraction and transmission of feature information.

Furthermore, by utilizing the CARAFE upsampling operator to enhance the upsampling process while simultaneously reducing FLOP by 3.9G, there was a 0.3% increase in mAP, leading to a notable improvement in accuracy. Additionally, integrating the SIoU algorithm to enhance the loss function resulted in a 0.6% increment in mAP value, contributing to better detection performance and convergence speed for the model. In comparison with the original model, it is evident that our proposed MCS-YOLO model has led to a significant improvement with an impressive 82.4% increase in mAP.

2) Comparison with the other lightweight YOLOv5s models

To further confirm the progressive nature of the proposed MGSEC3 module in lightweight modules. The C3 module in Table III was substituted with the lightweight MobileNetv3 module, ShuffleNetv2 module, and Ghostnet module. Respectively, we are named YOLOv5s-MobileNetv3, YOLOv5s-ShuffleNetv2, and YOLOv5s-Ghostnet. The experimental results show that Params and FLOPs are higher than other models when the MGSEC3 module is applied. However, mAP has significant advantages, being 3.5%, 2.1%, and 1.7% greater than the MobileNetv3 module, ShuffleNetv2 module, and Ghostnet module, respectively.

3) Comparison with the other attention mechanism network

Table IV is used to confirm the effectiveness of the SENet network architecture. We are replacing other attention mechanism network structures such as CBAM, ECA, CA, and SimAM network structures with SENet. The experimental results show that the mAP of SENet network structure is 0.5%, 0.9%, 1.1%, and 0.7% higher than CBAM, ECA, CA, and SimAM, respectively. SENet can significantly improve detection performance.

F. Comparison With Other State-of-the-art Detectors

We ran comparative experiments on our dataset to further analyze the detection performance of the suggested algorithm. The MCS-YOLO algorithm is compared with other popular object detection techniques. Each detection technique was tested in a consistent experimental context using identical data from both the training and test sets. Our system attained an accuracy of 82.4% on our dataset when compared with other algorithms, as demonstrated in Table V. Our approach outperforms the SSD model with a 22.7% greater mAP while still having lower parameters and computational complexity. Our model surpasses the other four YOLOv5 model structures with a higher mAP. Specifically, our model's mAP is 0.8% higher than YOLOv5x. And 6.8% and 13.0% higher than YOLOv7 and YOLOv7-tiny, respectively. Our model outperforms both the YOLOv4 and YOLOv4-tiny models with a higher mAP of 9.0% and 12.1%, respectively. Similarly, our model outperforms the Faster R-CNN two-stage object detection technique by achieving an accuracy that is 10.8% higher. The suggested technique demonstrates strong performance in defect detection in cylindrical coated lithium batteries after a thorough evaluation of several object detection algorithms.

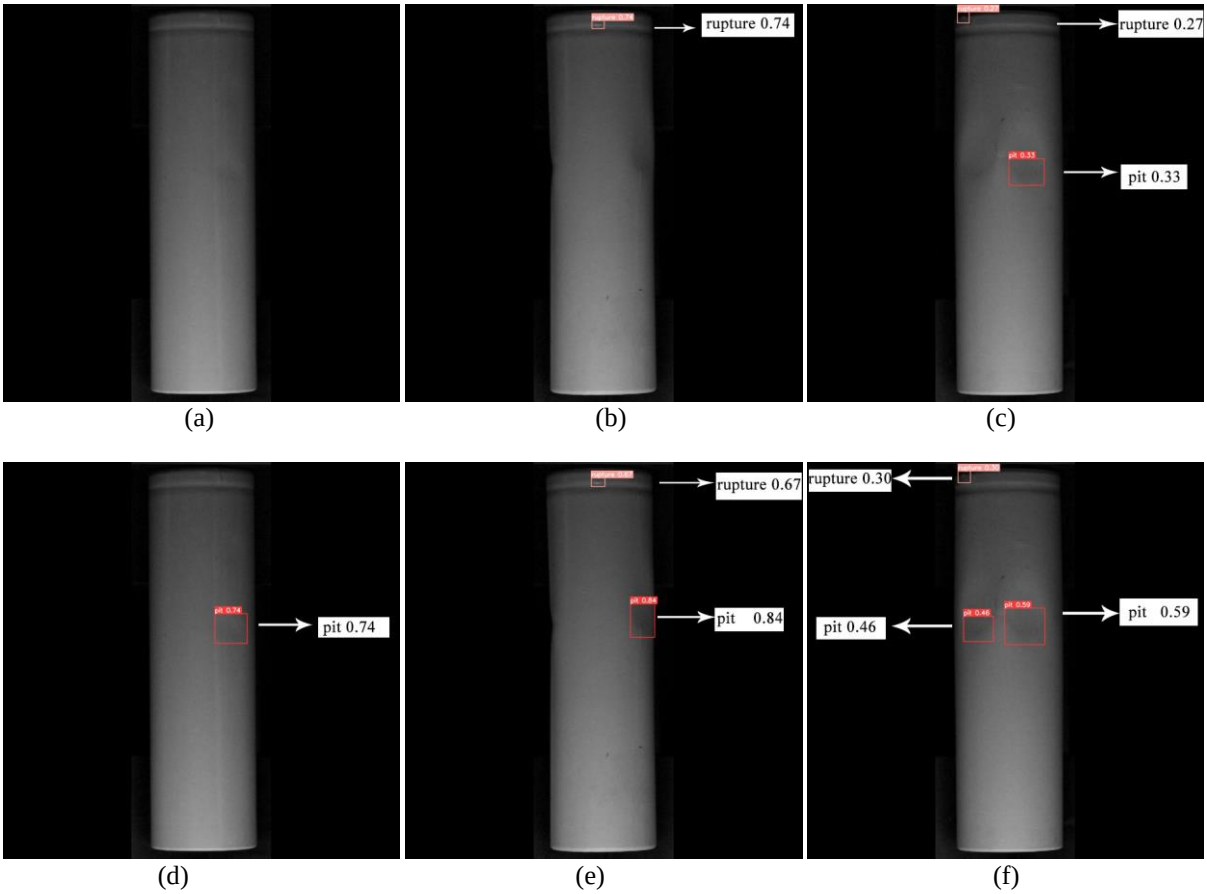

Fig. 11. Comparison of YOLOv5s and MCS-YOLO detection results

TABLE V
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS

Model	Size	m*AP*	Params	FLOPs
SSD	640x640	59.7	41.1	387.0
YOLOv5s	640x640	80.1	7.0	15.8
YOLOv5m	640x640	80.9	20.8	47.9
YOLOv5l	640x640	81.3	46.1	107.7
YOLOv5x	640x640	81.6	86.2	203.8
YOLOv4	640x640	73.4	6.3	97.0
YOLOv4-tiny	640x640	70.3	5.8	73.2
YOLOv7	640x640	75.6	37.2	105.1
YOLOv7-tiny	640x640	69.4	6.0	13.2
Faster R-CNN	640x640	71.6	108.0	128.3
MCS-YOLO	640x640	82.4	5.3	12.0

V.CONCLUSION

In this paper, considering that YOLOv5s has high computational load and memory requirements and is challenging to deploy on small devices, the C3 module in the Backbone network is improved, the upsampling algorithm of the Neck network is improved, and the loss function is also improved. Comparative validation was performed on the dataset using each improvement point to verify its effectiveness. In our future work, we will continue to explore the problem of defect detection in cylindrical coated lithium batteries, especially conducting more research on improving the detection effect to improve its practicality.

REFERENCES

[1] X. R. Zeng, "Research and Implementation of Visual Detection Algorithm for Battery Surface Defects," M.S. thesis, Dept. Information Science. Eng., Shenyang University of Technology., Liaoning, China, 2018.

[2] Z. C. Kang, "Research on Online Detection Method of Cylindrical Lithium Battery Surface Defects," M.S. thesis, Dept. Information Science. Eng., Shenyang University of Technology., Liaoning, 2019.

[3] Y. Li, "Research on the detection method of circumferential surface rupture of cylindrical coated lithium battery," M.S. thesis, Dept. Information Science. Eng., Shenyang University of Technology., Liaoning, China, 2020.

[4] C. X. Liu, "Research on detection method of circumferential surface defect of cylindrical lithium battery," M.S. thesis, Dept. Information Science. Eng., Shenyang University of Technology., Liaoning, China, 2022.

[5] S. T. Guo, "Research on Appearance Defect Detection Method of Cylindrical Lithium Battery," Ph.D. dissertation, Dept. Information Science. Eng., Shenyang University of Technology., Liaoning, China, 2022.

[6] X. Y. Feng, "Recognition and classification of surface defects of cylindrical lithium battery steel shell based on deep learning," M.S. thesis, Dept. Eng., Hefei University of Technology., Anhui, China, 2020.

[7] J. Tian, H. Hu, H. J. Zhou, R. Zhou, " Surface Defect Detection of Cylindrical Lithium Battery Based on Improved YOLOv4," Mechanical engineer, no. 3, pp. 16-18, 2023.

[8] H. B. Xu, "Research on Visual Detection of Surface Defects in Lithium Battery Cases Based on Deep Learning," M.S. thesis, Dept. Eng., Yanshan University., Hebei, China, 2021.

[9] Y. Q. Gui, L. S. Li, X. Mao, Q. Q. Wang, " Lithium battery defect detection method based on improved YOLOv4," Electronic measurement technology Electronic measurement technology, vol. 45, no. 15, pp. 144150, 2022.

[10] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[12] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," p. arXiv:1804.02767Accessed on: April 01, 2018. doi: 10.48550/arXi-v.1804.02767.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," p. arXiv:2004.10934Acces-sed on: April 01, 2020. doi: 10.48550/arXiv.2004.10934.

[14] Ultralytics.YOLOv5[EB/OL].(2020-06-3)[2021-4-15]. https://github.com/ultralytics/yolov5.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Computer VisionECCV2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp. 21–37, 2016.

[16] R. Girshick, J. Donahue, T. Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 580-587.

[17] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV),Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[18] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[19] J. Hu, L. Shen, G. Sun, "Squeeze-and-Excitation Networks," ArXiv, vol. abs/ 1709.01507, 2017.

[20] J. Q. Wang, K. Chen, R Xu, Zi. W. Liu, C. C. Loy, D. H. Lin, "Carafe: Content-aware reassembly of features" In 2019 The IEEE International Conference on Computer Vision (ICCV).

[21] Gevorgyan, Zhora, "SIoU Loss: More Powerful Learning for Bounding Box Regression," ArXiv abs/2205.12740 (2022).