Engineering Science and Technology, an International Journal 60 (2024) 101896

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full length article

LightYOLO: Lightweight model based on YOLOv8n for defect detection of ultrasonically welded wire terminations

Jianshu Xu [a,b], Lun Zhao [a,d,*], Yu Ren [a,c], Zhigang Li [b], Zeshan Abbas [a], Lan Zhang [a], Md Shafiqul Islam [d]

[a] School of Mechanical and Electrical Engineering, Yunnan Open University, Kunming, 650500, China
[b] School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China
[c] School of Biomedical Engineering, Shenzhen University, Shenzhen, 518060, China
[d] Department of Mechanical Engineering, Blekinge Institute of Technology, Karlskrona, 37179, Sweden

ARTICLE INFO

Keywords:
Ultrasonic metal welding
Deep learning
Object detection
Lightweight

ABSTRACT

Defect inspection of the surface in ultrasonically welded wire terminations is an important inspection procedure to ensure welding quality. However, the detection task of ultrasonic welding defects based on deep learning still faces the challenges of low detection accuracy and slow inference speed. Therefore, to solve the above problems, we propose a fast and effective lightweight detection model based on You Only Look Once v8 (YOLOv8n), named LightYOLO. Specifically, first, to achieve fast feature extraction, a Two-Convolution module with FasterNet block and Efficient multi-scale attention (CTFE) structures is introduced in the backbone network. Secondly, Group-Shuffle Convolution (GSConv) is used to construct the feature fusion structure of the neck, which enhances the fusion efficiency of multi-level features. Finally, an auxiliary head training method is introduced to extract shallow details of the network. To verify the effectiveness of the proposed method, we constructed a surface defect data set of ultrasonic welding wire terminals and conducted a series of experiments. The results of experiments show that the precision of LightYOLO is 93.4%, which is 3.5% higher than YOLOv8n(89.9%). In addition, the model size was reduced to 1/2 of the baseline model. LightYOLO shows the potential for rapid detection on edge computing devices. The source code and dataset for our project is accessible at https://github.com/JianshuXu/LightYOLO.

1. Introduction

Technology for ultrasonic welding of metals is recognized as a more stable and secure welding method, increasingly becoming the primary technique for connecting wires and terminals in the automotive industry [1]. Nevertheless, wire terminations subjected to ultrasonic welding may exhibit quality issues during the actual manufacturing process due to factors such as machinery or production processes. Common defects include rubber char, wire stain, and wire char, which can impair the electrical and mechanical properties of the wire termination joints. Consequently, automatically identifying and localizing these welding defects at wire termination connections is crucial for maintaining high-quality production standards. Traditionally, manufacturers of ultrasonically welded wire terminations have relied on manual visual inspections to identify product defects. While manual inspection offers flexibility in addressing various types and shapes of welding defects, it suffers from significant drawbacks, including subjectivity, insufficient detection efficacy and elevated labor costs. Compared to manual detection methods, automated defect detection systems demonstrate clear advantages. It is not only capable of adapting to challenging conditions but also sustains high precision and efficiency over extended periods.

Initially, several studies employed traditional computer vision algorithms for defect detection, which relies on manual analysis and extraction of defect features tailored to the vision inspection task. Subsequently, decisions are made based on rule-based experience or learning-based classifiers. For instance, Jian et al. [2] developed an enhanced detection algorithm specifically tailored for Mobile Phone Screen Glass (MPSG) defect identification and segmentation. This method employs a Contour-based Registration (CR) technique to generate template images, which aid in aligning MPSG images. Utilizing this registration, the subtraction and projection combination is applied to identify defects in the MPSG images. Similarly, Yuan et al. [3] introduced an advanced version of the Otsu method, termed Weighted Object Variance (WOV),

* Corresponding author at: School of Mechanical and Electrical Engineering, Yunnan Open University, Kunming, 650500, China.
E-mail addresses: JianshuXu@163.com (J. Xu), zhaolun@szpu.edu.cn (L. Zhao), renyu2023@email.szu.edu.cn (Y. Ren), li7275@163.com (Z. Li), abbasz@szpu.edu.cn (Z. Abbas), shafiqul.islam@bth.se (M.S. Islam).

https://doi.org/10.1016/j.jestch.2024.101896
Received 5 June 2024; Received in revised form 4 November 2024; Accepted 5 November 2024
Available online 22 November 2024

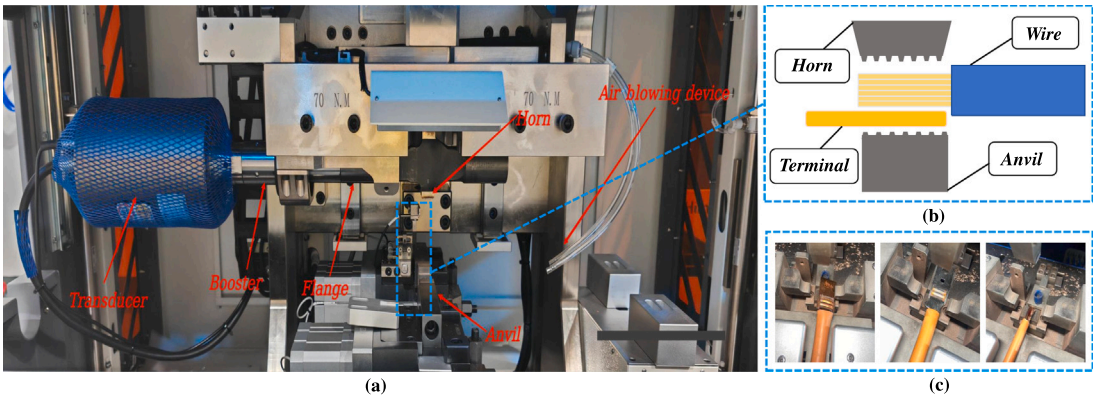


Fig. 1. (a) The ultrasonic welding equipment, (b) The details of ultrasonically welded wire terminations, (c) The products of ultrasonically welded.

aimed at identifying defects on product surfaces. Additionally, Cen et al. [4] proposed an automatic defect detection approach utilizing low-rank matrix reconstruction for Thin Film Transistor-Liquid Crystal Display (TFT-LCD) panels which offers the benefit of maintaining the edge and shape details of the defect area. Despite these advances, manual feature extraction and precise threshold setting remain tedious and susceptible to errors and necessitate a long development period to effectively accommodate a range of tasks involving complex surface defect features. Moreover, traditional computer vision algorithms typically exhibit limited generalization capabilities, making them less effective in handling new tasks or unknown scenarios.

Recently, the object detection technique utilizing Convolutional Neural Networks (CNNs) has been extensively applied in the surface defect detection of industrial products, which uses multiple layers of neural networks to grasp complex patterns and features from the data and does not need to manually extract defect features, making it very effective in identifying complex defects. For example, Zhang et al. [5] replaced the feature extraction network of Mask R-CNN with Efficient Net. At the same time, the Convolutional Block Attention Module(CBAM) is added to the mask branch to give the model a more comprehensive feature extraction capability. Zhang et al. [6] presented an improved YOLOv5 network that integrates the CBAM and C3 modules, replacing the existing C3 module. Additionally, the neck portion of the network incorporates a combination of BiFPN (Bidirectional Feature Pyramid Network) and Concat modules. These modifications enhance the network's ability to effectively extract and integrate features of defects. Nevertheless, there are existing limitations in employing CNNs for practical industrial defect detection. It is worth emphasizing here that fewer parameters and **fl**oating-point **op**eration**s** (FLOPs) are more important for defect detection models, as this facilitates the deployment of CNN to embedded devices with limited storage space and computational resources. Ma et al. [7] developed a lightweight detection model, leveraging the YOLOv4 framework, to address the specific needs of surface defect detection on aluminum strips. An attention mechanism featuring a dual-channel in parallel configuration was integrated to simultaneously capture channel and spatial attributes across multiple scales, thereby achieving a reduction in the number of model parameters. Tang et al. [8] proposed a novel lightweight model, light-PDD, for detecting defects in PCBs. The model adopted a pruned MobioeNev3 as its backbone, with an enhanced cross-stage partial(CSP) structure integrated into the neck section to eliminate superfluous parameters and streamline the model, thereby reducing computational complexity.

In the task of detecting surface defects on ultrasonic welding wire terminal ends, various types of defects exhibit significant differences in shape and size. Notably, defects such as wire broken, which are typically very small and randomly positioned, necessitate the use of high-resolution and high-precision detection technologies. Additionally, the background in the wire harness welding process usually comprises a multitude of complex equipment and tools. These elements often resemble the wire harness terminals in appearance, further complicates the distinction of targets from the background. Therefore, the detection model requires robust feature extraction capabilities to accurately extract target features from complex visual information. Furthermore, in industrial production, the detection of surface defects on ultrasonic welded wire harness terminals typically needs to be performed in real-time to facilitate prompt identification and correction of issues during the production process. Consequently, the detection algorithm must not only be highly accurate but also characterized by high efficiency and low latency. To solve these problems, and attain a balance between detection speed and accuracy. This study designs a lightweight ultrasonically welded wire terminations defect detection network (LightYOLO), which adopts the YOLOv8n network, a one-stage detection algorithm, as the baseline network. Firstly, the Faster Implementation of CSP Bottleneck with two convolutions(C2f) module is replaced by a CTFE module in the backbone section, enabling it to identify the defect location in the complex backgrounds and reduce parameters by reducing the redundant feature information between different channels. Secondly, GSConv was introduced to improve the detection speed, and the neck network is further lightweight through the VOV-GSCSP module. Finally, an auxiliary head is used in the middle layer of the network to enhance the network focus on shallow defect features. The improved network achieves superior detection accuracy with a reduced model size, effectively addressing the challenge that existing detection algorithms face in balancing detection speed and accuracy.

The principal contributions of this research are outlined as follows:

(1) A lightweight defect detection model based on YOLOv8n (LightYOLO) was proposed to meet the requirements of surface quality detection in ultrasonic welding of wire terminations.

(2) To enhance the multi-scale feature extraction capability of the network, the CTFE module was constructed by introducing Efficient Multi-Scale Attention module (EMA).

(3) To verify the efficacy of the proposed LightYOLO, an ultrasonically welded wire terminations defect data set (UWWT-Dataset) was constructed. The precision of the LightYOLO reaches 93.4%, and the detection speed reaches 176FPS.

2. Related methods

2.1. Ultrasonic welding of metals

Ultrasonic metal welding represents an effective method of solid-state welding. Due to its advantages of connecting a variety of dissimilar materials [9], environmental friendliness, and short process cycle time [10], it is widely used in high-precision connections. As the principal component of an ultrasonic metal welding machine, the ultrasonic vibration system is primarily composed of four key elements: an

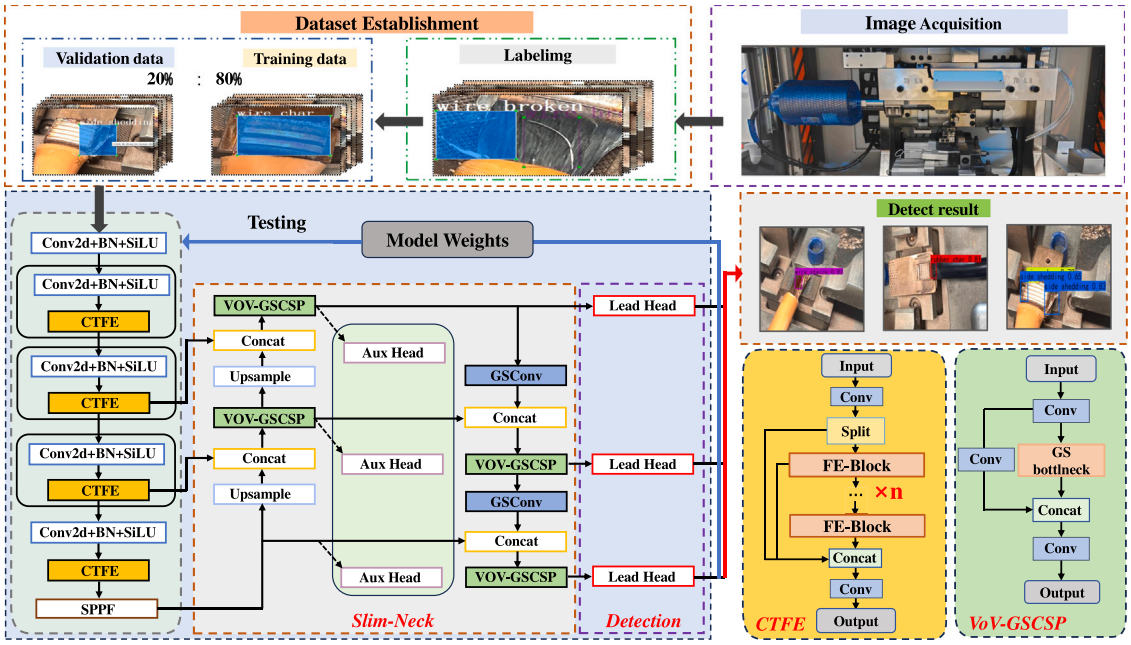


Fig. 2. The technical route and detail of the proposed architecture.

ultrasonic generator, a booster, a transducer, and a horn. First, the ultrasonic generator converts the power frequency current into ultrasonic frequency. Subsequently, this energy is transformed into mechanical vibration energy, which is then amplified by both a transducer and a booster. Finally, the amplified mechanical vibration energy is applied to the workpiece placed on the anvil through the horn, as shown in Fig. 1(a). In this way, the workpiece and the anvil move relative to each other, resulting in friction, local heating, and plastic deformation, as shown in Fig. 1(b). Ultimately, the connection of the workpiece to be welded is achieved [11], as shown in Fig. 1(c). However, due to uncontrollable factors such as material and tool conditions, the welding quality of the workpiece to be welded still changes greatly, so the consistency and quality of ultrasonic metal welding products cannot be guaranteed [12–14]. In addition, both the horn and the anvil are crucial to the quality of the welding [15,16]. But, as the number of welds increases, the patterns on the surfaces of the two tools will gradually wear out, creating defects.

2.2. Object detection

In recent years, object detection has emerged as a fundamental task in the field of computer vision, and its primary task involves identifying the category and location of targets within a given scene.[17]. Algorithms for object detection that employ a two-stage strategy, including R-CNN [18], Fast R-CNN [19], Faster R-CNN [20], and Mask R-CNN [21], adhere to the traditional concept of object detection. First, extracting features from candidate regions to generate Regions Of Interest (ROI), subsequently transferring these extracted features to the classifier, and the boundary box positions are determined by regression. This method performs classification and regression respectively and produces results with rich features and high detection accuracy. For example, Liu et al. [22] proposed a scheme for detecting defects on the surface of solar cells using Faster R-CNN, designed to facilitate the extraction of defect features across multiple scales. Meanwhile, the adaptive adjustment of anchoring was achieved using with Guided Anchoring Region Proposal Network (GA-RPN). However, the computational complexity causes the two-stage detection algorithm to require more inference time, which limits its application in scenarios that require higher detection speed. In contrast, algorithms for object detection that employ a one-stage strategy, such as YOLO [23–25] and SSD [26] directly implement the final results (category and location) [27]. In comparison to the two-stage detection algorithms, the one-stage detection algorithms provide faster inference speeds at the expense of sacrificing some accuracy. To ensure that the detection model maintains high stability and efficiency in actual industrial defect detection scenarios, the further enhancement of the single-stage object detector can effectively solve the detection of welding defects under complex backgrounds and diverse welding shapes and types.

2.3. Model lightweight

Deep learning-based object detection methods help develop intelligent industrial systems and reduce production losses. However, the significant computational costs require more expensive equipment, which remains a challenge for practical applications [28]. Therefore, the model needs to be lightweight. Currently, methods for miniaturizing neural networks can be divided into two categories: lightweight network structure design and model compression. The design of a lightweight network structure signifies the use of lightweight convolution operations, including group convolution and depth-separable convolution. For example, the SqueezeNet [29] reduces the computational complexity in the convolution process by using this method. Another method is model compression, which includes three techniques: knowledge distillation, quantization, and network pruning. Network pruning and quantization techniques reduce the number of network parameters and diminish overall network complexity by increasing the granularity of sparsity, allowing better utilization of the hardware of the computing platform [30]. However, although they can simplify the network structure, they usually require large pre-trained models for parameter compression and run the risk of accuracy degradation. The method of knowledge distillation can make the deep networks shallower and reduce the computational cost, but it also has its limitations. Although transfer models are relatively lightweight and efficient, additional computation and time are required to train and generate the transfer models. Designing more compact novel network structures is an emerging concept in network lightweight and acceleration, which no longer requires dedicated storage of pre-trained models like parameter compression methods, nor does it require fine-tuning to improve performance. Much current research on model lightweight focuses on compact networks. Therefore, this method is a more optimal choice.

3. LightYOLO defect detection algorithm for ultrasonically welded wire terminations

The developed LightYOLO, an iteration of the YOLOv8n network, incorporates an innovative CTFE structure. This novel design effectively minimizes the prevalent redundancy across feature maps in different channels, consequently enhancing the detection speed of the network. This section will explain in detail the CTFE structure, the lightweight strategy of the neck network and the optimization of the training method by adding the auxiliary head. Through the aforementioned design, the LightYOLO architecture achieves a reduction in the number of network parameters while simultaneously preserving a high level of detection performance. The technical route, along with details of the LightYOLO architecture, is depicted in Fig. 2.

3.1. Two-convolution module with FasterNet block and efficient multi-scale attention

In the defect detection task of ultrasonically welded wire terminal, the feature maps between different channels are highly similar, as shown in Fig. 3. This is due to the C2f module using more layer-hopping connections, the stacking of large bottleneck structures inevitably leads to a high redundancy of channel information and an increase in network parameters. This redundant channel information will increase additional computing costs and memory usage during feature extraction, resulting in higher delays and affecting the efficiency of the defect detection task of the ultrasonically welded wire terminal. To solve the above problems, some classic lightweight networks, such as MobileNet [31] and ShuffleNet [32] try to use depthwise separable convolution (DSC) [33] to extract spatial features to reduce the number of floating-point operations (FLOPs). MicroNet [34] further decomposes and sparsifies the network, thereby reducing the FLOPs to exceptionally low levels. Nevertheless, the above networks frequently incorporate additional data operations, such as pooling and shuffling, the execution times of which can substantially affect the performance of lightweight models. Chen et al. [35] point out that reducing FLOPs alone does not necessarily lead to the reduction of network latency. From (1), it is evident that the **fl**oating-point **op**erations per **s**econd (FLOPS) and FLOPs both exert an influence on latency.

$$Latency = \frac{FLOPs}{FLOPS} \tag{1}$$

Therefore, in the backbone section of the baseline network, a more lightweight CTFE was designed to replace the C2f to reduce additional computation, as shown in Fig. 4(a). The designed CTFE module enhances the gradient flow within the model through the incorporation of additional branches for gradient flow. Herein, the CBS module is comprised of Conv2d, BatchNorm2d, and SiLU activation functions. The fast and efficient block(FE-Block), as shown in Fig. 4(b), is mainly composed of partial convolution (PConv) as shown in Fig. 4(c), and pointwise convolution (PWConv). After the first PWConv operation, each feature input is normalized and activated using Batch Normalization (BN) [36] and the **re**ctified **l**inear **u**nit (ReLU). This helps to maintain a stable input distribution across layers, enhances the sparsity of the network, and thereby accelerates the convergence speed of the network. Finally, the incorporation of an EMA module [37] enables the model to more effectively recognize and distinguish between different classes or objects.

After extracting features through CBS, PConv only applies standard convolution to part of its channels, then utilizes PWConv to process information from the remaining channels. Notably, PConv significantly reduces FLOPs, as illustrated in (2), When the ratio is set to $r = 1/3$, the FLOPs for PConv are reduced to 1/9 times those of standard convolution. Meanwhile, Chen et al. [35] find that frequent memory accesses lead to low FLOPs. Thus, the relationship between FLOPS and the Amount of Memory Access (AMA) can be considered inversely proportional. As shown in (3), when the ratio is similarly set to $r = 1/3$, the FLOPS of PConv are $r = 1/3$ of those of standard convolution. Therefore, the latency of PConv is approximately $r = 1/3$ that of standard convolution.

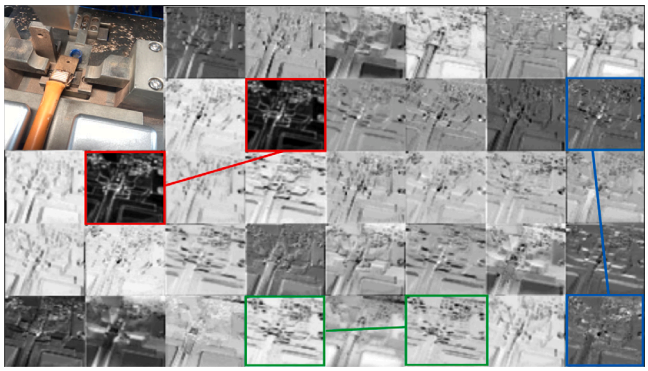


Fig. 3. Visualization of the middle layer feature map Image of the network with the top left image as input.

$$FLOPs_{PConv} = h \times w \times k^2 \times c_p^2 \tag{2}$$

$$FLOPS \propto \frac{1}{AMA} = \frac{1}{h \times w \times 2c_p + k^2 \times c_p^2} \approx \frac{1}{h \times w \times 2c_p} \tag{3}$$

The EMA module, as shown in Fig. 4(d), effectively integrates contextual information across various scales. This enhances the quality of representation for each pixel in the deep feature maps, thereby yielding more precise detection outcomes.EMA module first performs a grouping operation on the initial feature map $X \in R^{C \times H \times W}$, which is partitioned into G sub-features ($g_x = [X_0, X_i, \ldots, X_{G-1}]$) along the channel dimension to extract comprehensive semantic details. Meanwhile, the EMA module references the design concept of parallel substructures for Coordinated Attention (CA) [38]. Parallel substructures facilitate networks to circumvent extensive linear processing and significant depth. The above process can be formulated as

$$x_h, x_w = AvgPool\left[(g_x)_h, (g_x)_w\right] \tag{4}$$

$$x_1 = Conv_{3\times3}(g_x) \tag{5}$$

Where $Conv_{3\times3}(\cdot)$ indicates the convolution with the kernel size 3×3, which effectively captures spatial patterns of pixels and their neighboring pixels, aiding in the extraction of local features from the image. Subsequently, through the concat operation, two branches are vertically concatenated, and a convolution is performed on the concatenated feature map using a 1×1 convolution kernel. This operation allows for the integration of information across different channels while preserving spatial dimensions. Subsequently, the outputs of the convolution are decomposed into two vectors, each processed through a sigmoid function to adjust and optimize the probability distribution of the data. Finally, the information from the two parallel 1×1 branches is fused through a simple multiplication operation to obtain a new feature representation.

During the cross-spatial learning process, the output from 1×1 branch is initially subjected to Group Normalization (GN). Subsequently, both 1×1 and 3×3 branches undergo global average pooling, followed by the application of the Softmax function. This facilitates the extraction of spatial information from the feature map and aligns with the linear transformations. At this time, the spatial attention maps x_{11} and x_{21} are obtained. Additionally, the outputs from 1×1 and 3×3 branches will be transformed into their respective dimensional shapes, x_{12} and x_{22}. The features are subsequently combined with previously

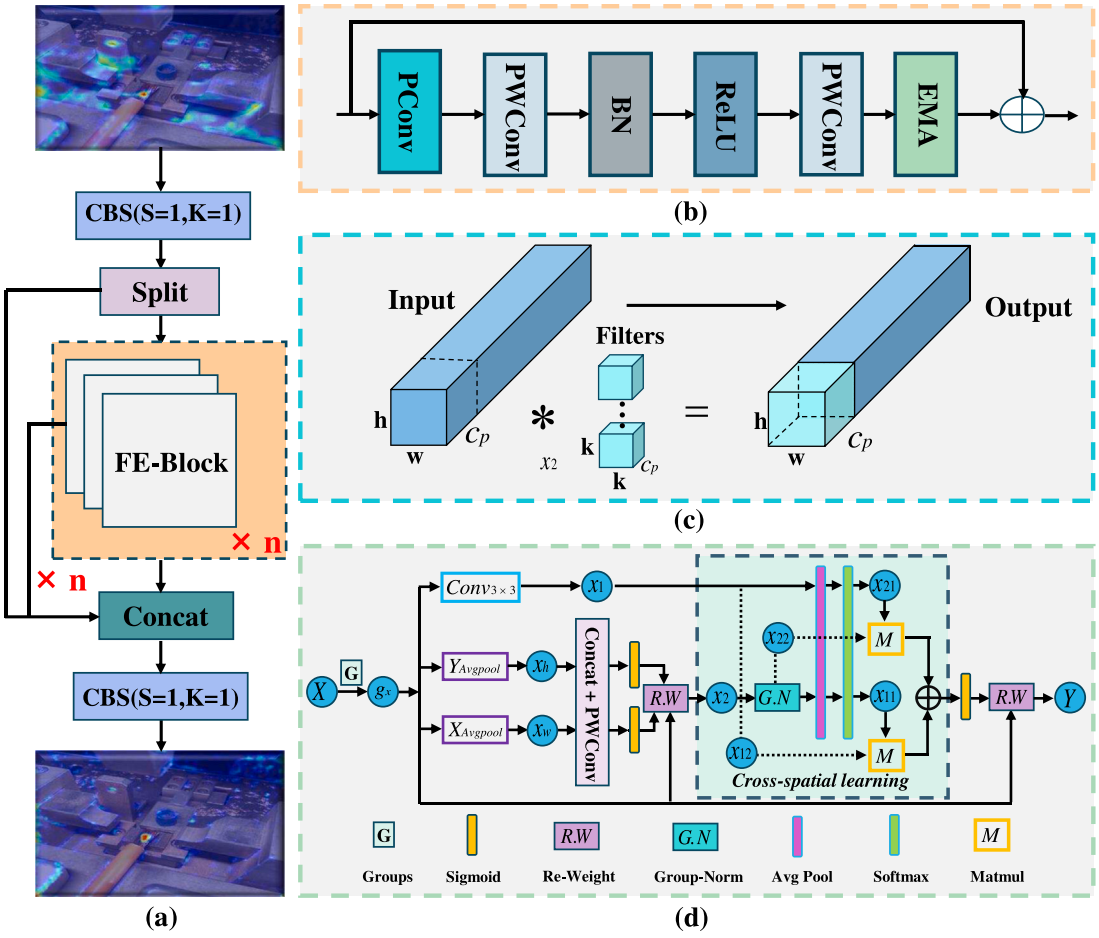


Fig. 4. The structure of (a) CTFE, (b) FE-Block, (c) PConv, and (d) EMA module.

obtained feature maps, and after aggregating the two attention weight values, they are processed through a Sigmoid function. Ultimately, a multiplication with g_x is performed to produce the final output.

$$x_{11}, x_{21} = softmax(AvgPool(x_1, GN(x_2))) \tag{6}$$

$$Y = g_x \times (\sigma(x_{11} \otimes x_{12}) + \sigma(x_{21} \otimes x_{22})) \tag{7}$$

Where σ denotes the function of sigmoid. The resulting output of EMA module maintains the identical dimensions as X, making it both efficient and suitable for integration into modern architectures.

3.2. Slim-Neck to reduce computational cost

For the surface defect detection task of ultrasonically welded wire terminals, the detection accuracy is as significant as the detection speed. The use of Depthwise separable convolution (DSC) can achieve the purpose of a lightweight network, but it cannot guarantee the accuracy of detection. This can be primarily attributed to the complexity of defects in ultrasonically welded wire terminals., including shape, texture, color, etc. DSC usually uses a small convolution kernel to independently perform convolution operations on each channel of the input, which significantly reduces the computational load. However, this approach constrains the capacity of the network to learn more complex features. Conversely, standard convolution (SC) takes into account all channels of the input simultaneously and preserves the relationships between channels to the greatest extent. However, this approach significantly increases the computational cost compared to DSC. Therefore, a convolution operation that combines the respective advantages of SC and DSC is highly anticipated.

As shown in Fig. 5(a), the Group-shuffle convolution (GSConv) [39] concatenates the feature maps produced by the SC and the DSC. Subsequently, it employs a channel shuffling strategy to enhance the interchange of information between features. Fig. 5(b) and Fig. 5(c) illustrate the computational processes of SC and DSC, respectively. When the spatial dimensions of the feature map are reduced and the channels are expanded, SC consistently preserves the hidden connections between each channel to the greatest extent possible (DSC completely disrupts these connections). Through this method, avoiding the loss of some semantic information, making the feature map of GSConv closer to the feature mapping generated by SC. The FLOPs of SC, DSC, and GSConv are shown in (8). Where $W \times H$ denote the size of the output feature map; K_1, K_2 represent the size of the convolution kernel; C_1 and C_2 are the number of channels in the input and output feature map respectively.

$$\begin{aligned} FLOPs_{SC} &= W \times H \times K_1 \times K_2 \times C_1 \times C_2 \\ FLOPs_{DSC} &= W \times H \times K_1 \times K_2 \times 1 \times C_2 \\ FLOPs_{GSConv} &= W \times H \times K_1 \times K_2 \times \frac{C_2}{2} \times (C_1 + 1) \end{aligned} \tag{8}$$

In addition, since CTFE is used in the feature extraction network, there is less repeated information when the feature map reaches the feature fusion network, and no compression is required. Therefore, GSConv is employed solely during the feature fusion stage to ensure the effectiveness and efficiency of data flow, even in the context of deeper network layers. To further reduce inference time and maintain

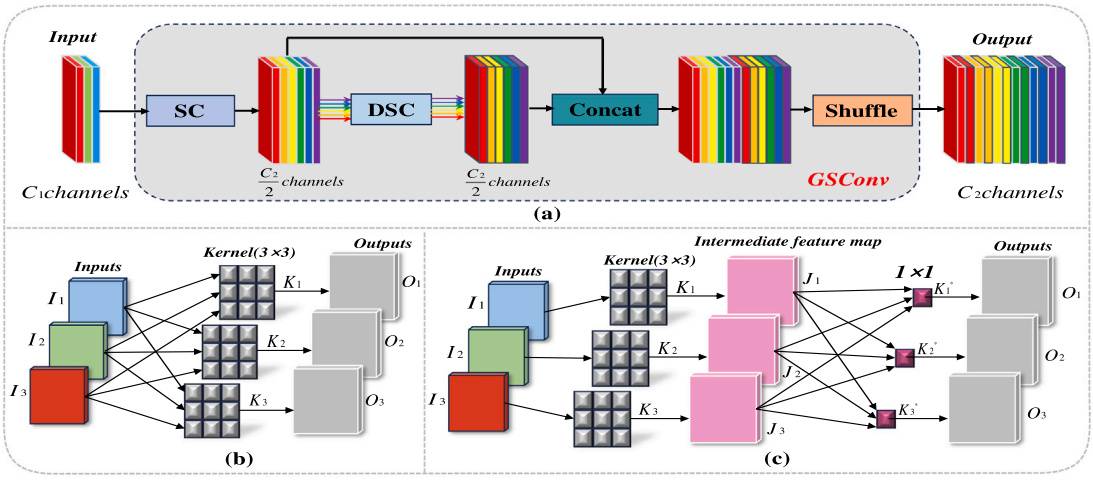


Fig. 5. (a) The structure of the GSConv module. The calculation process of the (b) SC and the (c) DSC.

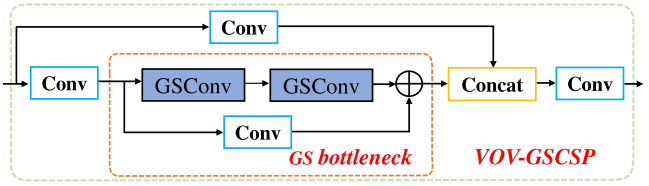


Fig. 6. The structure of the GS bottleneck module and VOV-GSCSP module.

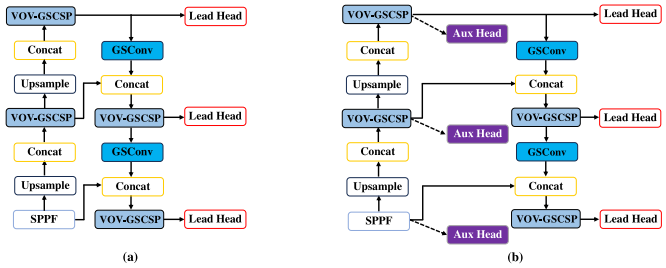


Fig. 7. (a) Incorporating auxiliary detection heads at intermediate network layers (9, 12, 15) aids in the early capture of target details. (b) With final detentions occurring at deeper layers (15, 18, 21) using abstract features for precise classification and localization.

detection accuracy, the VoV-GSCSP module, depicted in Fig. 6, has been introduced. This module includes a GS bottleneck module based on GSConv, designed to replace the C2f module in the neck. The architecture of the VoV-GSCSP, which utilizes one-shot aggregation can not only leverage the multi-receptive fields of Dense Net to represent different features but also overcome the inefficiency of dense connections.

3.3. Add extra auxiliary head in the middle layers

Although using pooling or convolutional layers for down-sampling can decrease the size of feature maps and reduce both data volume and computational complexity, this method results in a lowered resolution of the feature maps. This reduction can lead to the loss of critical detail information, especially for small defects in the original image, such as wire broken. Therefore, to enhance the detection capabilities for small targets, this study integrates an auxiliary detection head at the initial layers, as shown in Fig. 7, effectively leveraging its high spatial resolution. This strategy ensures that the rich feature information of the shallow layers, such as texture and boundary details, is effectively utilized, significantly enhancing the overall performance of the network. Moreover, the introduction of the auxiliary detection head aids in faster convergence of the detection network during training. This benefit primarily arises from their role as additional sources of gradients, which facilitates the optimization of shallow layer weight adjustments, especially in the early stages of network training.

The auxiliary detection head was introduced in the intermediate layers of the network, the loss function of the model $LOSS$ includes four components:classification loss $LOSS_{BCE}$, bounding box loss $LOSS_{Bbox}$, dynamic feature learning loss $LOSS_{DFL}$ and auxiliary loss $LOSS_{Aux}$. Among these, the classification loss uses binary cross-entropy loss to calculate the difference between the predicted and actual categories, as detailed in (9). The bounding box loss employs CIoU loss to measure the discrepancy between the predicted and actual bounding boxes, as specified in (10). The dynamic feature learning loss is used to improve the efficiency with which the model utilizes features, especially in the detailed prediction of bounding boxes. (11) represents the formula for calculating the total loss. In this context, γ represents the weight ratio of the auxiliary loss, used to adjust the weight of the auxiliary function within the total loss, thereby controlling the impact of the loss function on model training. In this study, the weight ratio of the auxiliary loss is set to 0.25 to ensure that it does not dominate the training process. Instead, it aids the primary loss in achieving better generalization effects and maintains a stable flow of gradients throughout the training process, particularly to prevent the issue of gradient vanishing in deep network training.

$$LOSS_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)] \tag{9}$$

$$CIoU = 1 - IoU + \frac{\rho^2(b, b_{gt})}{c^2} + \alpha v \tag{10}$$

$$LOSS = LOSS_{Bbox} + LOSS_{BCE} + LOSS_{DFL} + \gamma \times LOSS_{Aux} \tag{11}$$

Further, the auxiliary head and the lead head are separated, and then the respective prediction results and ground truth are used to perform the label assignment. In this paper, the "label assigner" mechanism is introduced to allocate labels. Guided by lead head prediction, hierarchical labels from coarse to fine are generated, respectively, for auxiliary head and lead head learning. The soft labels it generates better represent the distribution and correlation between the original data and the target data. This kind of learning can be viewed as generalized residual learning in that by allowing the shallower auxiliary head to directly learn the information that the lead head has learned, the Lead head will be better able to focus on learning the remaining information

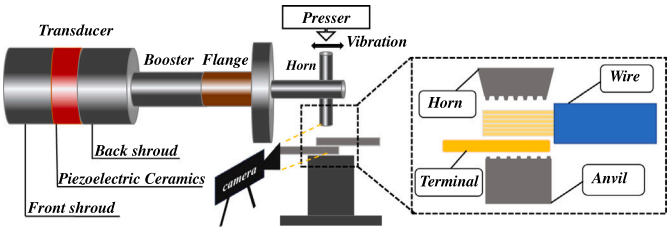


Fig. 8. Acquisition of Ultrasonically welded wire terminations dataset.

that has not been learned. It is worth noting that the auxiliary head is only used during the training stage of the network and not during inference. This method helps reduce the calculation amount of the network during the inference process and improves the inference efficiency.

Overall, by incorporating auxiliary detection heads in the intermediate layers of the network and using auxiliary loss to guide the adjustment of weights in shallow layers, significant improvements can be made in the early learning performance of the model. This promotes more efficient feature learning, thereby enhancing the overall training efficiency and accuracy of the model. It is important to note that the auxiliary detection heads are only used during the training phase and are not involved during inference. This helps reduce the computational load during the inference stage and increases inference speed.

4. Experiments

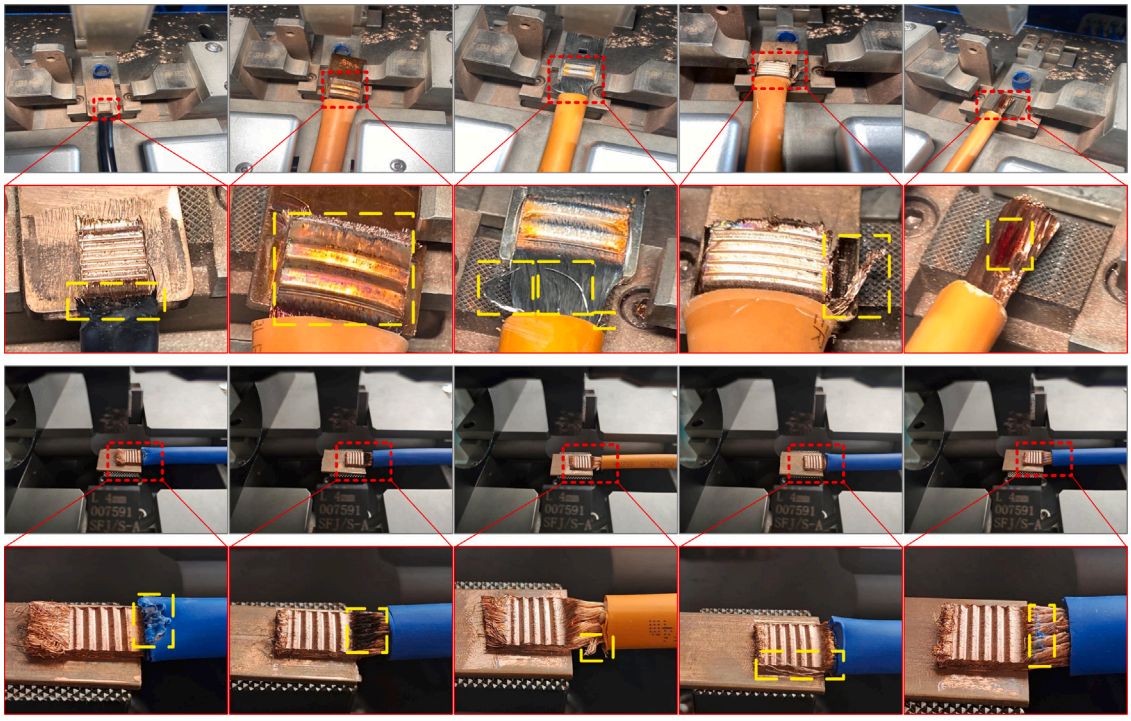
4.1. Datasets

As shown in Fig. 8, ultrasonic welding equipment is usually designed to be aligned with the workpiece from the vertical direction. Place the workpiece on the fixture below while the welding head applies pressure from above so that the welding part completely covers the welding area during the welding process. It is difficult to get a clear image of the welded area from the front or top. Additionally, the horizontal corners of the weld area are often open, making the side view a more feasible perspective for image capture. It can be seen from the side angle that during the ultrasonic welding process, welding defects (such as broken wires and side peeling) tend to become more obvious on the side of the workpiece. To verify that the methods proposed in this paper will improve the detection efficiency of the model, an ultrasonic welding wire terminal defect dataset was captured on the side in an actual industrial environment, named the UWWT-Dataset. There are five defect types, including Rubber Char (RC), Wire Char (WC), Wire Broken (WB), Side Shedding (SS) and Wire Stain (WS), As shown in Fig. 9, the defective part is enlarged. The dataset contains 635 samples, to ensure the robustness and generalization ability of the model, image flipping and the addition of image noise were used to augment datasets. The augmented datasets comprise 2231 defect images split into training and validation sets in an 8:2 ratio. The distribution of defects in the datasets is as follows: 346 RC images, 631 WC images, 703 WB images, 396 SS images, and 155 WS images. Labeling annotation software is used to annotate datasets.

In addition, we use the publicly available dataset NEU-DET as a standard benchmark for model performance to verify the robustness and generalization of the proposed model. NEU-DET is a surface defect dataset released by Northeastern University, which collected six typical defects of hot-rolled steel strips, including Crazing (Cr), Patches (Pa), Inclusion (In), Pitted Surface (PS), Rolled-in Scale (RS), and Scratches (Sc), the details are shown in Fig. 10. Each class of defects contains 300 samples.

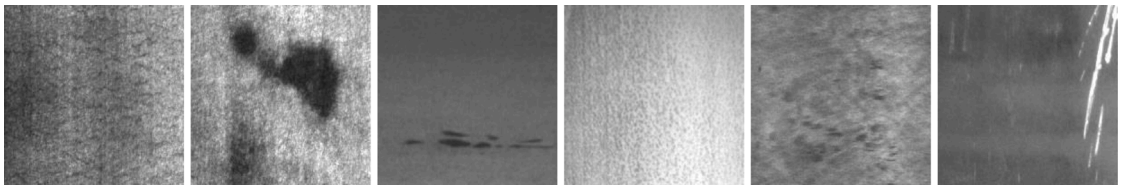
4.2. The experiments environment and evaluation metrics

The experiments were conducted on a system equipped with an Intel i5-12400F processor and an NVIDIA GeForce RTX 3060 graphics card, featuring 16 GB of memory. The version of PyTorch used was 2.1.0 and the version of CUDA was 12.2. Some hyper-parameter settings during the training process are as follows: the batch size was 16,



(a) RC (b) WC (c) WB (d) SS (e) WS

Fig. 9. Five types defects of the ultrasonically welded wire terminations surface: (a) rubber char, (b) wire char, (c) wire broken, (d) side shedding, (e) wire stains. The defect details are in the yellow dotted box.



(a) Cr (b) Pa (c) In (d) PS (e) RS (f) Sc

Fig. 10. Six types of defects of the hot-rolled steel strips surface: (a) crazing, (b) patches, (c) inclusion, (d) pitted surface, (e) rolled-in scale, (f) scratches.

Table 1
Performance comparison results of different object detection networks on UWWT.

Model	Precision(%)	Recall(%)	mAP(%)	Model Size(MB)	Parameters(M)	FLOPs(G)	FPS
Faster R-CNN	88.5	87.2	87.9	125.7	136.8	401.8	23.0
SSD	92.1	79.5	82.3	92.6	26.3	62.7	43.0
RetinaNet [41]	91.6	83.3	87.1	139.0	36.4	164.9	42.0
YOLOv5s	92.4	85.1	88.3	27.1	9.1	23.8	106.0
YOLOv8n	89.9	83.5	86.6	11.6	3.0	8.3	168.0
YOLOv9c	91.2	87.3	90.3	253.6	32.6	119.3	153.0
RT-DETR [42]	88.9	86.1	87.3	69.3	20.2	58.3	64.0
LightYOLO	93.4	86.4	88.6	6.4	2.4	6.4	176.0

the training epoch was 300, the weight decay was 0.0005, the initial learning rate was 0.01, and cosine annealing was used as the learning rate scheduling strategy. Meanwhile, the optimization process utilized Stochastic Gradient Descent (SGD) with Nesterov momentum, with the momentum parameter set to 0.937.

To evaluate the performance of the proposed network, the evaluation metrics employed include precision, recall, and mean Average Precision (mAP). mAP is the mean of AP and is an indicator to evaluate the overall performance of an object detection algorithm on multiple categories. The Average Precision (AP) indicates the accuracy of a model for a specific category, determined by calculating the area under the Precision–Recall (P–R) curve. It can reflect the accuracy and recall of the algorithm for different categories. Generally speaking, a higher mAP value indicates that the algorithm has better average performance across multiple categories. The following equations describe the detailed interpretation of these indicators:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$AP = \frac{\sum P}{N} \tag{14}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP(n) \tag{15}$$

Here, TP, FP, and FN denote true positives, false positives, and false negatives, respectively. N represents the total number of classes, and $AP(n)$ signifies the average precision.

4.3. The results and analysis of experiments

4.3.1. Comparison of different object detection network

To validate the performance of the proposed model, we plan to compare it with other leading object detection models available, such as the one-stage detection models YOLOv5s and YOLOv9c [40], and the two-stage detection model Faster R-CNN. This comparison will be based on the same training strategy, to ensure the fairness and accuracy of the comparison results. The results from the comparative analyses are displayed in Table 1.

As evident from the table, among the detection networks mentioned, LightYOLO achieves the satisfactory performance with an mAP of 88.6%, exceeding Faster R-CNN (87.9% mAP) by 0.7% and exceeding SSD (82.3% mAP) 6.3%, RetinaNet (87.1% mAP) 1.5%, YOLOv5s (88.3% mAP) 0.3%, RT-DETR (87.3% mAP) 1.3%, although YOLOv9c has a slightly higher mAP than LightYOLO but at the same time, the model size of YOLOv9c is nearly 40 times larger than LightYOLO. It is worth noting that LightYOLO is 2% higher than the baseline network YOLOv8n (86.6% mAP). As shown in the bar chart in Fig. 11, LightYOLO demonstrates improved detection accuracy, recall, and mAP compared to YOLOv8n. Faster R-CNN achieves the best recall performance because its two-stage design allows the model to select candidate bounding boxes more accurately. However, this improvement in recall comes at the cost of significantly reduced model inference speed. Among all the networks compared, it has the lowest Frames Per Second (FPS) is the lowest at only 23, which is insufficient for defect detection in ultrasonic welded wire terminals. Likewise, SSD and RetinaNet perform poorly in model inference speed due to their high computational complexity and large model size, making them unsuitable for deployment on edge computing devices. The inference speed of the YOLO series network is sufficient to meet the needs of actual application scenarios. The YOLOv5s network exhibits a higher mAP than the YOLOv8n network; however, its performance on other indicators lags behind that of YOLOv8n. This suggests that there is still room for improvement in the YOLOv8n network. Moreover, compared to YOLOv9c and RT-DETR, LightYOLO achieves a favorable balance among several critical metrics, including model size, computational complexity, and the number of computational parameters. This balance is crucial for resource-constrained edge computing devices. In Fig. 12, the performance of different models in the task of detecting surface defects in ultrasonic welding of wire terminals is illustrated. Each column represents a typical defect type, while each row corresponds to the results of a specific detection model. It can be observed that for the wire broken defect, which involves small target areas, Faster R-CNN, SSD, and RetinaNet all exhibit missed detection. Notably, RetinaNet misclassifies SS as WS. In contrast, the remaining detection models successfully identified all defects. Among them, LightYOLO demonstrated outstanding performance while maintaining minimal parameter count and computational complexity.

To verify the robust performance of LightYOLO, we conducted comparative experiments with different object detection networks on the NEU-DET dataset, and the experimental results are shown in Table 2. Upon analysis of the data presented in the table, it is evident that LightYOLO exhibits remarkable performance metrics, specifically in terms of Precision (P), Recall (R), and mAP, scoring 69.9%, 72.2%, and 77.4%, respectively. Its mAP score is particularly noteworthy, ranking second only to YOLOv9c and achieving the second-highest performance level. LightYOLO achieves this performance with a significantly lower

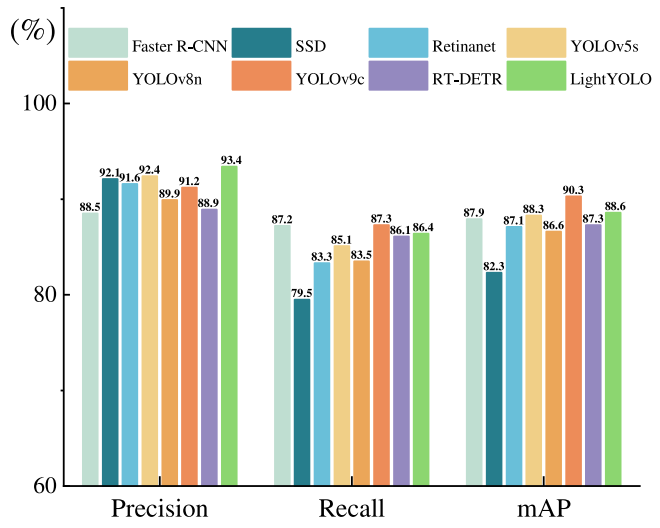


Fig. 11. Comparison of the Precision, Recall, and mAP of each object detect networks.

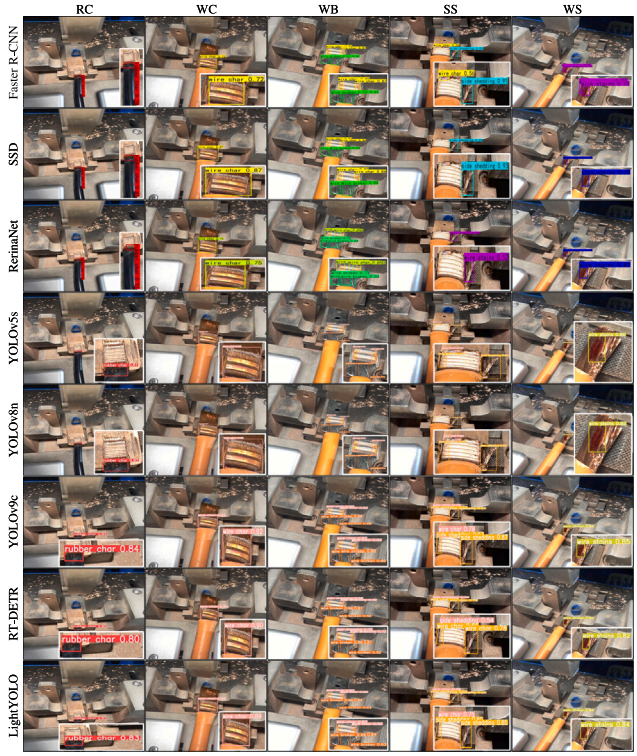


Fig. 12. Results detected by different object detection networks on UWWT-Dataset, the bottom right corner shows the detection results in detail.

number of FLOPs compared to YOLOv9c, approximately one-twentieth, indicating a superior balance between accuracy and detection speed. This balance makes LightYOLO particularly suitable for edge computing devices, as it achieves satisfactory results while imposing minimal computational demands on the hardware. Such efficiency can primarily be attributed to the reduction in computational load facilitated by employing PConv techniques in CTFE, along with integrating an EMA module to enhance defect recognition focus. Fig. 13 provides a comparative illustration of confusion matrices. Fig. 13(a) and Fig. 13(b) juxtapose YOLOv8n and LightYOLO. Within these matrices, columns represent predicted categories, rows signify actual categories, and values along the diagonal indicate the accuracy of model predictions. Observing

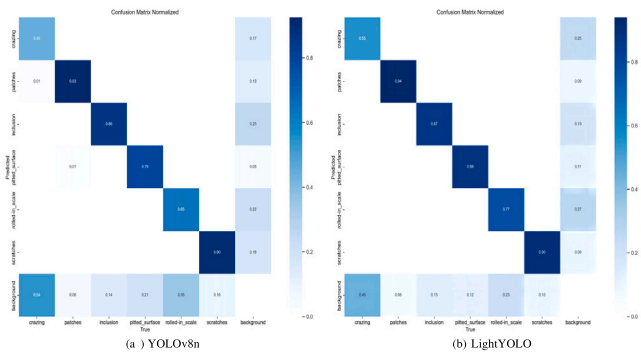


Fig. 13. Confusion matrix of different models. (a) and (b) show that YOLOv8n and LightYOLO, respectively.

the values along the diagonals of the two matrices, it can be seen that LightYOLO shows consistent increases. This indicates that LightYOLO achieves higher prediction accuracy across all defect categories, thereby once again verifying the effectiveness of LightYOLO.

Fig. 14 illustrates the visual detection results of various sample defects using object detection networks that exhibit balanced performance across multiple metrics. It is evident that for CR-type defects, YOLOv8n fails to detect defects, while RT-DETR exhibits instances of missed detections. For Pa-type defects, both YOLOv9c and RT-DETR encounter false detections. Additionally, YOLOv8n demonstrates missed detections when identifying In-type and Rs-type defects. In comparison to other methods, LightYOLO provides more accurate defect localization and higher reliability, with fewer instances of missed and false detections.

Additionally, to validate the performance of LightYOLO in real-world scenarios under challenges such as occlusion, lighting variations, and scale changes, we conducted experiments on the MS COCO dataset. The proposed model was tested and compared with other models, with the results presented in Table 3. Analysis of the data reveals that LightYOLO outperforms other object detection models in handling real-world tasks across diverse environments, angles, and backgrounds. Although it exhibits slightly lower detection accuracy compared to YOLOv9c and RT-DETR, LightYOLO demonstrates significant advantages in terms of parameter count and computational complexity, indicating its substantial potential for practical applications.

In general, compared with other object detection networks, LightYOLO has achieved better performance, which can the meet precision and speed requirements necessary for defect detection tasks of ultrasonically welded wire terminations to the greatest extent.

4.3.2. Comparative experiments on different attention modules

Additional tests were conducted to further validate that the introduction of the EMA module will enable the model to focus on the important parts of the input feature maps. The Two-Convolution module with FasterNet block (CTF) is combined with several different attention modules. such as Sim Attention module (SimA) [43], Triple Attention module (TA) [44] and SK Attention module (SKA) [45], and the results of the experiments are displayed in Table 4.

As evidenced by the table, significant variations were observed in network performance when different attention modules were used in combination with Fast block. Without the addition of the attention module, the mAP stands at merely 85.9%. Among them, when using the Fast block combined with the EMA module, the recall reached 86.4%, and the precision also reached the best performance of 93.4%. At the same time, the combination with EMA module (88.6%) performs better than the other three attention module combinations: SimA module (86.4%), TA module (87.4%) and SKA module (86.2%). This is because the EMA module, in processing input data, not only focuses on local

Table 2
Performance comparison results of different object detection networks on NEU-DET.

Model	P(%)	R(%)	mAP(%)							Para(M)	FLOPs(G)
			Cr	In	Pa	PS	RS	Sc	All		
Faster R-CNN	67.4	73.2	47.2	85.3	94.7	86.2	62.3	69.5	76.6	136.8	401.8
SSD	68.9	71.6	48.8	83.6	95.1	84.6	64.1	65.2	73.4	26.3	62.7
RetinaNet	65.3	68.3	47.8	75.3	95.8	82.9	71.2	35.2	66.8	36.4	164.9
YOLOv5s	68.5	67.9	39.6	84.3	94.7	84.6	60.3	91.6	74.6	9.1	23.8
YOLOv8n	67.5	68.8	36.4	82.6	88.5	87.6	56.6	92.4	74.0	3.0	8.3
YOLOv9c	71.2	70.2	40.8	86.8	91.1	88.2	51.2	87.7	77.9	32.6	119.3
RT-DETR	74.0	65.5	30.8	80.4	91.8	84.2	57.2	89.9	72.4	20.2	58.3
LightYOLO	69.9	72.2	42.3	89.1	92.9	85.2	62.1	92.8	77.4	2.4	6.4

Table 3
Comparative experimental results of different models on MS COCO.

Model	Size(pixels)	mAP50 val(%)	Parameters(M)	FLOPs(G)
Faster R-CNN	640 × 640	55.7	136.8	401.8
SSD	640 × 640	50.4	26.3	62.7
RetinaNet	640 × 640	59.1	37.7	167.3
YOLOv5s	640 × 640	56.8	7.2	16.5
YOLOv8n	640 × 640	59.8	3.2	8.7
YOLOv9c	640 × 640	70.2	25.3	102.1
RT-DETR	640 × 640	63.8	20.6	61.2
LightYOLO	640 × 640	61.2	2.6	6.7

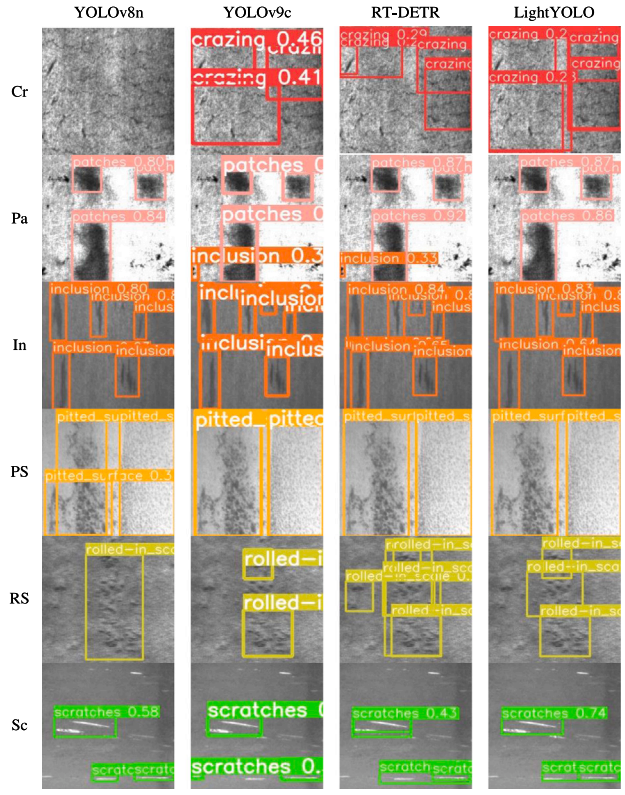


Fig. 14. Results detected by different object detection networks, each depicted in distinct colors to facilitate differentiation.

information but also analyzes global data characteristics to optimize the channel weights within each parallel processing branch. Furthermore, by further integrating cross-dimensional information, the capability of the model to capture pixel-level pairwise relationships in the data is enhanced. From the aforementioned analysis, it can be observed that the introduction of the attention module can enhance the performance of the model without adversely affecting its efficiency.

From the Table 4, it is evident that the integration of EMA module and CTF leads to an increase in both the parameters and FLOPs of the

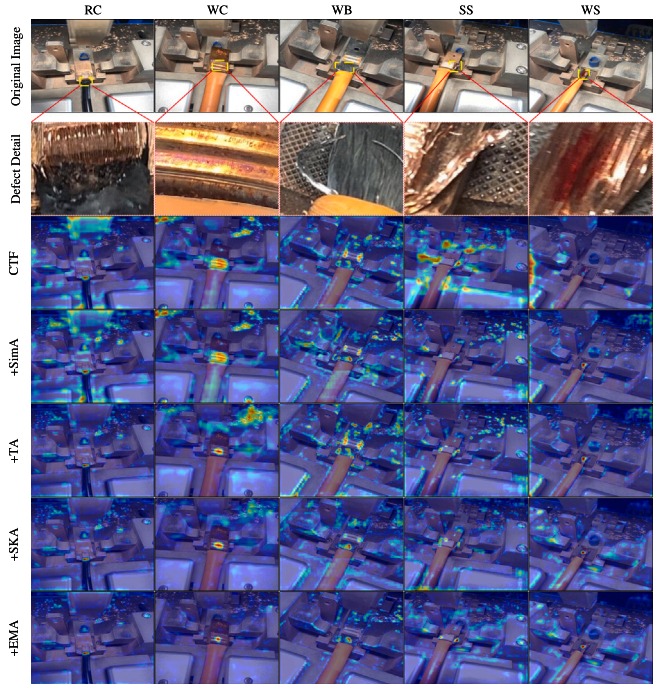
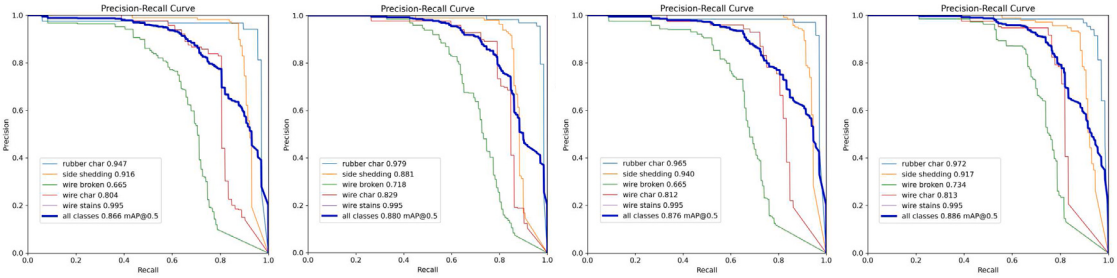


Fig. 15. The Grad-CAM heat map of CTF and after combining different attention modules with CTF.

Table 4
Comparative experiments on different attention modules.

Module	Precision(%)	Recall(%)	mAP(%)	Parameters(M)	FLOPs(G)
CTF	92.6	79.4	85.9	2.3	6.3
+SimA	91.4	81.4	86.4	2.3	6.3
+TA	92.2	83.4	87.4	2.3	6.3
+SKA	90.4	82.3	86.2	2.3	6.3
+EMA	93.4	86.4	88.6	2.4	6.4

model. This increase primarily stems from the substantial additional computational steps introduced by the EMA module. Specifically, the inclusion of various types of convolutional layers, such as 1 × 1 and 3 × 3 convolutions, has significantly increased the parameter count and computational demands. Additionally, multi-scale pooling operations and the matrix multiplication required for calculating attention weights have introduced an additional computational burden. Group Normalization operations not only increase the required normalization parameters but also correspondingly enhance the computational load. In contrast, other attention mechanisms such as SimA module primarily employ linear computational methods (calculating mean or variance), which essentially do not increase the parameters. TA module and SKA module primarily involve limited convolutional and pooling operations, which, although leading to a moderate increase in parameters and computational demands, are relatively minor compared to those introduced by EMA module.



(a) Baseline (b) +CTFE (c) +CTFE+SN (d) +CTFE+SN+AH

Fig. 16. P–R curves of ablation study for diverse parts: (a) Baseline, (b) CTFE, (c) Slim-Neck, (d) Aux-Head.

Table 5
Ablation experiments.

Baseline	CTFE	Slim-Neck	Aux-Head	mAP(%)						Para (M)	FPS
				RC	WC	WB	SS	WS	ALL		
✓				94.7	91.6	66.5	80.4	99.5	86.6	3.0	168.0
✓	✓			97.9	88.1	71.8	82.9	99.5	88.0	2.6	171.0
✓	✓	✓		96.5	94.0	66.5	81.2	99.5	87.6	2.4	176.0
✓	✓	✓	✓	97.2	91.7	73.4	81.3	99.5	88.6	2.4	176.0

To qualitatively analyze the superiority of CTFE structure in welding defect feature extraction, CTF and the combination of CTF with different attention modules were visualized on UWWT-Dataset through Grad CAM, as shown in Fig. 15. Observation of the heatmap reveals that, without the addition of an attention module, the network fails to accurately locate defect areas and erroneously focuses on some background regions. In contrast, the CTF architecture with an attention module demonstrates better performance, capable of more precisely identifying defective areas while effectively minimizing attention to irrelevant regions to some extent. Among them, the combination of EMA module and CTF achieved outstanding performance. As revealed by the last row of the figure, it can fully focus on the welding defect area in the complex background and diminish the disturbances due to complex backdrops.

4.3.3. Ablation experiments

To investigate the effects of CTFE, slim neck, and auxiliary head on LightYOLO performance, we conducted ablation experiments and the results are presented in Table 5. When the C2f module is replaced with the CTFE module, the mAP is 88.0%, the parameters are 2.6M, and the FPS is 171. For the baseline network, mAP increases by 1.4% due to the addition of the CTFE module. The number of parameters is decreased by 0.4M, while the inference speed of the model improves by 3 FPS. This shows that while CTFE introduces the EMA module to improve model detection accuracy, it can still reduce channel view information redundancy through Faster Block, thereby reducing computational complexity and accelerating model inference. When the slim neck structures are further used, the model achieves a mAP of 87.6%, which is 1% greater than that of the baseline network, and the parameters are further reduced to 2.4 M. This is because GSConv has the advantages of both SC and DSC, and the VoV-GSCSP structure can reduce the number of parameters in the network. Finally, by implementing the auxiliary head training strategy, the network demonstrates a 2% enhancement in mAP in contrast to the baseline network. This is because using the auxiliary head to guide the soft labels will help the lead head extract residual information from the object, thereby elevating the detection performance of the network without increasing the calculations in the inference stage.

The precision–recall (P–R) curve is a graphical representation in the coordinate system comprising precision and recall, with the enclosed region representing the average precision(AP), as shown in Fig. 16. The P–R curve demonstrates that optimizing the architecture and training strategies of the baseline network can lead to improved performance in defect detection. From Fig. 16(a), it is evident that the detection effect of the baseline network for different categories of defects and the mean mAP. Fig. 16(b) shows the effect when C2f is replaced by CTFE, observational analysis reveals an increase in the area under the green curve depicted in the graph. The defect type represented by the green curve is wire is broken, which is characterized by a changeable shape. This indicates that the attention module can effectively enhance the capability of the network to extract features of irregular welding defects. When GSConv and VoV-GSCSP structures are further used in the network, as shown in Fig. 16(c), although the mAP of the network is reduced by 0.4%, the number of parameters and computational complexity of the network is significantly reduced. In addition, when the auxiliary head is added to the middle layer of the network, it can be observed from Fig. 16(d) that the area surrounded by LightYOLO is larger than that surrounded by YOLOv8n, and the detection effect of different types of defects is improved.

5. Conclusions

For the task of surface defect detection of ultrasonically welded wire terminals, we propose a defect detection network based on YOLOv8n named LightYOLO. The CTFE module is designed to be applied in the backbone network to reduce redundant information in the feature maps and enhancing the focus on the defect locations. To make the network lighter and achieve a balance between detection speed and accuracy, we introduced GSConv in the neck network. Finally, to extract shallow details of the network, an auxiliary head training method is introduced. The proposed LightYOLO network is compared with other object detection networks. The final experimental results show that LightYOLO performs well in the detection task of ultrasonically welded wire terminal surface defects, with mAP reaching 88.6% and FPS reaching 176. Moreover, the public domain dataset NEU-DET is used to standardize the testing environment to validate the generalization performance of the LightYOLO. Experimental results indicate that among all comparative networks, LightYOLO achieved outstanding performance. Overall, the LightYOLO model shows certain performance improvements in the detection accuracy and detection speed of ultrasonically welded wire terminal surface defects, making it more suitable for actual industrial scenarios. Next, we will deploy this algorithm into actual scenarios to perform real-time defect detection on the wire terminal after ultrasonic welding. This will help improve the quality of wire terminal connection products and ensure driving safety.

CRediT authorship contribution statement

Jianshu Xu: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Lun Zhao:** Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis. **Yu Ren:** Writing – review & editing, Investigation, Formal analysis, Data curation. **Zhigang Li:** Writing – review & editing, Visualization, Supervision, Formal analysis. **Zeshan Abbas:** Writing – review & editing, Visualization, Investigation, Formal analysis. **Lan Zhang:** Writing – review & editing, Validation, Formal analysis, Data curation. **Md Shafiqul Islam:** Writing – review & editing, Validation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 12104324); Scientific Research Startup Fund for Shenzhen High-Caliber Personnel of SZPU (No. 6022310046K); SZPU-Newpower Ultrasonic welding R and D (602331009PQ); Postdoctoral Later-stage Foundation Project of Shenzhen Polytechnic University (6023271014K1).

References

[1] X. Cheng, K. Yang, J. Wang, W. Xiao, S. Huang, Ultrasonic system and ultrasonic metal welding performance: A status review, J. Manuf. Process. (2022) URL https://api.semanticscholar.org/CorpusID:253465031.

[2] C. Jian, J. Gao, Y. Ao, Automatic surface defect detection for mobile phone screen glass based on machine vision, Appl. Soft Comput. 52 (2017) 348–358, URL https://api.semanticscholar.org/CorpusID:205708981.

[3] X. Yuan, L. Wu, Q. Peng, An improved Otsu method using the weighted object variance for defect detection, Appl. Surf. Sci. 349 (2015) 472–484, URL https://api.semanticscholar.org/CorpusID:137557202.

[4] Y. Cen, R. Zhao, L.-H. Cen, L. hong Cui, Z. Miao, Z. Wei, Defect inspection for TFT-LCD images based on the low-rank matrix reconstruction, Neurocomputing 149 (2015) 1206–1215, URL https://api.semanticscholar.org/CorpusID:6165334.

[5] C. Zhang, B. quan Yu, W. Wang, Steel surface defect detection based on improved MASK RCNN, in: 2022 IEEE 8th International Conference on Computer and Communications, ICCC, 2022, pp. 2176–2181, URL https://api.semanticscholar.org/CorpusID:257647617.

[6] S. Zhang, M. He, Z. Zhong, D. Zhu, An industrial interference-resistant gear defect detection method through improved YOLOv5 network using attention mechanism and feature fusion, Measurement (2023) URL https://api.semanticscholar.org/CorpusID:260789800.

[7] Z. Ma, Y. bo Li, M. Huang, Q. Huang, J. Cheng, S. Tang, A lightweight detector based on attention mechanism for aluminum strip surface defect detection, Comput. Ind. 136 (2022) 103585, URL https://api.semanticscholar.org/CorpusID:245498116.

[8] J. Tang, Z. Wang, H. Zhang, H. Li, P. Wu, N. Zeng, A lightweight surface defect detection framework combined with dual-domain attention mechanism, Expert Syst. Appl. 238 (2023) 121726, URL https://api.semanticscholar.org/CorpusID:263251653.

[9] W. Cai, B. Kang, S.J. Hu, Ultrasonic welding of lithium-ion batteries, 2017, URL https://api.semanticscholar.org/CorpusID:136206051.

[10] Y. Meng, C. Shao, Physics-informed ensemble learning for online joint strength prediction in ultrasonic metal welding, Mech. Syst. Signal Process. (2022) URL https://api.semanticscholar.org/CorpusID:250190698.

[11] I. Balz, E.A. Raad, E. Rosenthal, R. Lohoff, A. Schiebahn, U. Reisgen, M. Vorländer, Process monitoring of ultrasonic metal welding of battery tabs using external sensor data, J. Adv. Joining Process. (2020) URL https://api.semanticscholar.org/CorpusID:213702310.

[12] L. Nong, C. Shao, T.H. Kim, S.J. Hu, Improving process robustness in ultrasonic metal welding of lithium-ion batteries, J. Manuf. Syst. (2018) URL https://api.semanticscholar.org/CorpusID:115811964.

[13] Z. Ma, Y. Zhang, Characterization of multilayer ultrasonic welding based on the online monitoring of sonotrode displacement, J. Manuf. Process. 54 (2020) 138–147, URL https://api.semanticscholar.org/CorpusID:216245718.

[14] X. Shi, L. Li, S. Yu, L. Yun, Process monitoring in ultrasonic metal welding of lithium batteries by power signals, J. Manuf. Sci. Eng. (2021) URL https://api.semanticscholar.org/CorpusID:242138265.

[15] C. Shao, W. Guo, T.H. Kim, S.J. Hu, J.P. Spicer, J. Abell, Characterization and monitoring of tool wear in ultrasonic metal welding, 2014, URL https://api.semanticscholar.org/CorpusID:264237697.

[16] Q. Nazir, C. Shao, Online tool condition monitoring for ultrasonic metal welding via sensor fusion and machine learning, J. Manuf. Process. 62 (2021) 806–816, URL https://api.semanticscholar.org/CorpusID:234364521.

[17] M. Yuan, C. Zhang, Z. Wang, H. Liu, G. Pan, H. Tang, Trainable spiking-YOLO for low-latency and high-performance object detection, Neural Netw.: Off. J. Int. Neural Netw. Soc. 172 (2023) 106092, URL https://api.semanticscholar.org/CorpusID:266962726.

[18] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 580–587, URL https://api.semanticscholar.org/CorpusID:215827080.

[19] R.B. Girshick, Fast R-CNN, 2015, URL https://api.semanticscholar.org/CorpusID:206770307.

[20] S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2015) 1137–1149, URL https://api.semanticscholar.org/CorpusID:10328909.

[21] K. He, G. Gkioxari, P. Dollár, R.B. Girshick, Mask R-CNN, 2017, URL https://api.semanticscholar.org/CorpusID:54465873.

[22] L. Liu, Y. Zhu, M.R.U. Rahman, P. Zhao, H. Chen, Surface defect detection of solar cells based on feature pyramid network and GA-faster-RCNN, in: 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence, CCHI, 2019, pp. 292–297, URL https://api.semanticscholar.org/CorpusID:208208461.

[23] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 779–788, URL https://api.semanticscholar.org/CorpusID:206594738.

[24] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 6517–6525, URL https://api.semanticscholar.org/CorpusID:786357.

[25] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, ArXiv, arXiv:1804.02767. https://api.semanticscholar.org/CorpusID:4714433.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, in: European Conference on Computer Vision, 2015, URL https://api.semanticscholar.org/CorpusID:2141740.

[27] C. Zhao, X. Shu, X. Yan, X. Zuo, F. Zhu, RDD-YOLO: A modified YOLO for detection of steel surface defects, Measurement (2023) URL https://api.semanticscholar.org/CorpusID:257764025.

[28] H.M. Ahmad, A. Rahimi, Deep learning methods for object detection in smart manufacturing: A survey, J. Manuf. Syst. (2022) URL https://api.semanticscholar.org/CorpusID:250161229.

[29] F.N. Iandola, M.W. Moskewicz, K. Ashraf, S. Han, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $< 1MB$ model size, 2016, ArXiv, arXiv:1602.07360. https://api.semanticscholar.org/CorpusID:14136028.

[30] W. Guo, X. Li, Z. Shen, A lightweight residual network based on improved knowledge transfer and quantized distillation for cross-domain fault diagnosis of rolling bearings, Expert Syst. Appl. (2023) URL https://api.semanticscholar.org/CorpusID:266642027.

[31] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, ArXiv, arXiv:1704.04861. https://api.semanticscholar.org/CorpusID:12670695.

[32] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 6848–6856, URL https://api.semanticscholar.org/CorpusID:24982157.

[33] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1800–1807, URL https://api.semanticscholar.org/CorpusID:2375110.

[34] Y. Li, Y. Chen, X. Dai, D. Chen, M. Liu, L. Yuan, Z. Liu, L. Zhang, N. Vasconcelos, MicroNet: Improving image recognition with extremely low FLOPs, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 458–467, URL https://api.semanticscholar.org/CorpusID:236987171.

[35] J. Chen, S. hong Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, S.-H.G. Chan, Run, don't walk: Chasing higher FLOPS for faster neural networks, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 12021–12031, URL https://api.semanticscholar.org/CorpusID:257378655.

[36] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, ArXiv, arXiv:1502.03167. https://api.semanticscholar.org/CorpusID:5808102.

[37] D. Ouyang, S. He, J. Zhan, H. Guo, Z. Huang, M. Luo, G.-L. Zhang, Efficient multi-scale attention module with cross-spatial learning, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2023, pp. 1–5, URL https://api.semanticscholar.org/CorpusID:258535361.

[38] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 13708–13717, URL https://api.semanticscholar.org/CorpusID:232110359.

[39] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, Q. Ren, Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles, 2022, ArXiv, arXiv:2206.02424. https://api.semanticscholar.org/CorpusID:249394561.

[40] C.-Y. Wang, I.-H. Yeh, H. Liao, YOLOv9: Learning what you want to learn using programmable gradient information, 2024, ArXiv, arXiv:2402.13616. https://api.semanticscholar.org/CorpusID:267770251.

[41] T.-Y. Lin, P. Goyal, R.B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2999–3007, URL https://api.semanticscholar.org/CorpusID:47252984.

[42] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, Y. Liu, DETRs beat YOLOs on real-time object detection, 2023, ArXiv, arXiv:2304.08069. https://api.semanticscholar.org/CorpusID:258179840.

[43] L. Yang, R.-Y. Zhang, L. Li, X. Xie, SimAM: A simple, parameter-free attention module for convolutional neural networks, in: International Conference on Machine Learning, 2021, URL https://api.semanticscholar.org/CorpusID:235825945.

[44] D. Misra, T. Nalamada, A.U. Arasanipalai, Q. Hou, Rotate to attend: Convolutional triplet attention module, in: 2021 IEEE Winter Conference on Applications of Computer Vision, WACV, 2020, pp. 3138–3147, URL https://api.semanticscholar.org/CorpusID:222177028.

[45] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 510–519, URL https://api.semanticscholar.org/CorpusID:80628366.