Journal of King Saud University - Computer and Information Sciences 36 (2024) 101936

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

EMB-YOLO: Dataset, method and benchmark for electric meter box defect detection

Zhiyong Liu [a], Yong Li [a,*], Feng Shuang [a], Zhongmou Huang [b], Ruichen Wang [a]

[a] Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China
[b] Chongzuo Power Supply Bureau of Guangxi Power Grid Co., Ltd, Chongzuo, China

ARTICLE INFO

Keywords:
Defect detection
YOLO
Power inspection
Meter Box

ABSTRACT

The electric meter box is a terminal device with a large number in the power grid. It may cause electrical hazards and property loss if damaged. Inspection of electricity meter boxes still relies on manual inspection with low efficiency and low automation. But image-based automated inspection is also limited by equipment battery and insufficient computing power, which makes the inspection system in urgent need of efficient model. However, lightweight model may reduce model robustness and be susceptible to interference from complex backgrounds due to insufficient feature extraction. Meanwhile, there are no publicly available datasets for electric meter boxes at present. To address the above issues, we firstly constructed a dataset, named EMB-11. After that, we improved the YOLOv7-tiny to design a novel model for electric meter box defect detection, named EMB-YOLO. In EMB-YOLO, we proposed the Big Kernel ShuffleBlock which can increase the effective receptive field and reduce the model parameters. Additionally, we proposed ELAN-CBAM to enhance the robustness of the model and reduce the interference of background noise. Finally, we constructed RepBSB based on the idea of structural reparameterization to reduce the size of the trained model. Compared to YOLOv7-tiny, the size of EMB-YOLO is only 4.82 Mb, which is reduced by 20.3 %. The detection speed is 343 frames/s, which is increased by 14.3 %. Most importantly, mAP can reach 82.8 %, which is increased by 3.5 %, reaching the SOTA level.

1. Introduction

Electric meter boxes are widely distributed in the fields of electricity consumption, serving as crucial measuring equipment for charging by power grid companies. In order to ensure the stability of the final link from the transmission line to the house, researchers have made efforts in various aspects (Duan et al., 2022); including adding various sensors inside the meter box. However, there are currently very few researchers conducting research from the perspective of safeguarding the meter box equipment itself. In fact, due to their exposure to the outdoors and prolonged service, a majority of meter boxes exhibit varying degrees of damage, resulting in consistently high rates of repairs and customer complaints. A damaged meter box poses serious risks, including electrical paralysis, accelerated aging of internal wiring and sensors, leakage, fire, and other hazardous consequences. In addition, the study of electricity theft is currently an important topic in power research (Qi et al., 2022; Zidi et al., 2023; Takiddin et al., 2023; Takiddin et al., 2021), and damage to the meter box can also easily facilitate electricity theft, causing property losses to power grid companies (Xia et al., 2022).

Currently, the detection of defects in electric meter boxes heavily relies on manual inspection, presenting challenges due to the decentralized distribution of these boxes. The verification process involves numerous technical points, making manual inspection prone to omissions and resulting in inefficiencies. Moreover, the high level of expertise required for verification poses difficulties for ordinary personnel. The repair requests from users often lack professional knowledge to accurately identify the type of damage, which makes it difficult for technical personnel to provide targeted on-site repairs, and also leads to low maintenance efficiency.

In the early days, researchers developed electronic devices, such as tamper detection devices, to detect whether the meter box is intruded or tampered (McLaughlin et al., 2013). With the progress and development of technology, power grid companies have begun to use image acquisition devices such as drones and fixed cameras to capture images of power equipment. The increasing number of images has led to the application of image recognition technology for defect identification in power equipment (Xu et al., 2022).

It is also necessary to introduce image recognition technology into

* Corresponding author.
E-mail address: yongli@gxu.edu.cn (Y. Li).

https://doi.org/10.1016/j.jksuci.2024.101936
Received 9 September 2023; Received in revised form 17 January 2024; Accepted 17 January 2024
Available online 22 January 2024

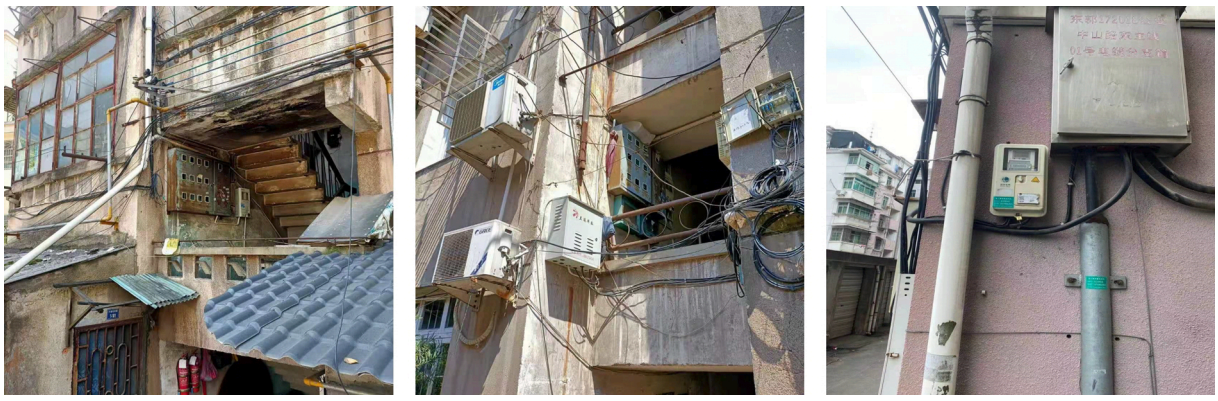


Fig. 1. Electricity meter box under complex background.

the inspection process of electric meter boxes. Specifically, in rural or sparsely populated areas, meter boxes usually hang under the eaves of individual buildings. For these areas with sparse distribution of meter boxes, drones can be used for inspection. Conversely, in densely populated areas like residential buildings or shopping malls, where meter boxes are usually centrally managed at the stairwell entrance of each floor or in a separate small area, robots can be used to conduct inspections layer by layer. Moreover, in order to improve repair efficiency, users or inspection personnel can perform self-inspection by mobile phones and other portable devices.

In recent years, deep learning has achieved outstanding results in the field of image recognition, and various industries have also applied it to defect detection. Introducing image defect detection technology that based on deep learning into the field of meter box defect detection can greatly improve the efficiency of meter box inspection. The main task of power equipment defect recognition based on deep learning is to identify the defects in images, and make classification, location and semantics understanding to the images. After years of development, more and more researchers are applying convolutional neural models in deep learning to power equipment defect recognition. Compared to traditional image recognition algorithms, convolutional neural models have better detection accuracy and stronger generalization ability. There are two main types of convolutional neural model for classical object detection, namely two-stage detector based on region candidates, i.e., Region-based Convolutional Neural Network (R-CNN) (Fast, 2015) and Region-based Fully Convolutional Network (R-FCN) (Dai et al., 2016), one-stage detector based on regression, i.e., You Only Look Once (YOLO) (Bochkovskiy et al., 2004) and Single Shot MultiBox Detector (SSD) (Liu et al., 2016).

Despite the increasing automation of electric power inspection, Edge devices such as unmanned aerial vehicles (UAVs), robots and handheld computers have become crucial tools for inspection. However, Edge device is constrained by computing power and battery. Therefore, the one-stage detector with low computing power requirements and high detection efficiency is more suitable for scenarios involving edge devices. For the one-stage detectors, the YOLO series eliminated the mechanism of object region recommendation and achieved satisfactory detection efficiency. They directly performed object category classification and bounding box regression on a complete image through convolutional neural models. Therefore, it is highly favored in current engineering applications. Nevertheless, while the one-stage detector is faster than the two-stage detector, most of the one-stage detectors are still unable to obtain satisfactory result on Edge device. On the one hand, it is because the current power inspection equipment has relatively low computational power, and the one-stage detector still cannot achieve satisfactory detection speed. Furthermore, the detection accuracy of the one-stage detector is insufficient. The lightweight version of the one-stage detector basically retained main architecture of the original version while using smaller convolutional kernels, fewer convolutional layers, and removing many feature extraction branches. This design can significantly reduce the number of parameters and improve computational speed, but insufficient feature extraction and insufficient feature fusion result in a significant decrease in detection accuracy. Especially, the current detection algorithms have not targeted optimization for different detection scenarios in power inspection. In the scenario of defect detection for electric meter boxes, whether in urban or rural areas, there are relatively complex backgrounds and many sundries that interfere with the detection, as shown in Fig. 1. Enabling deep learning models to accurately locate and detect areas of interest, reduce noise interference, and improve model robustness is another key factor that could affect detection accuracy. Consequently, achieving a balance between accuracy and speed on one-stage detectors becomes a critical research focus for algorithmic applications on edge devices.

However, the current shortage of samples is a common problem in the field of electric power inspection, especially in existing meter box defect detection. Factors such as high difficulty in data collection and the lack of integration of image-based techniques contribute to the absence of an available public dataset.

To address the above problems, we have improved the YOLOv7-tiny to design a novel model named EMB-YOLO. In EMB-YOLO, we have proposed three modules, i.e., Big Kernel ShuffleBlock (BSB), Efficient layer aggregation network with Convolutional block attention module (ELAN-CBAM) and Reparameterization Big Kernel ShuffleBlock (RepBSB). These enhancements collectively render the entire model more lightweight while simultaneously achieving higher detection accuracy. The primary contributions of this paper are outlined as follows:

1) **Dataset Construction:** We have constructed a new dataset for electricity meter box named EMB-11. In addition, we have utilized the current advanced lightweight object detection algorithms and our proposed algorithm to establish a comprehensive performance benchmark on this dataset.
2) **Model Design:** We have designed a model named EMB-YOLO. This model is specifically tailored for the defect detection scenario of electric meter boxes. In this model, we design BSB and RepBSB to reduce model size. The parameter of EMB-YOLO is 20.3 % less than the benchmark model, while achieving a reasoning speed 343 FPS. Thus, the proposed model exhibits faster speed and lower computational power requirements.
3) **Innovative Modules:** A self-attention layer aggregation model named ELAN-CBAM and an efficient big kernel convolutional layer aggregation model named BSB-ELAN has constructed. These modules improve the performance of model in feature extraction, resulting in a 3.5 % increase in mAP_{50} and an improved positioning performance of the benchmark model, leading to a 5.5 % increase in $mAP_{50:95}$.

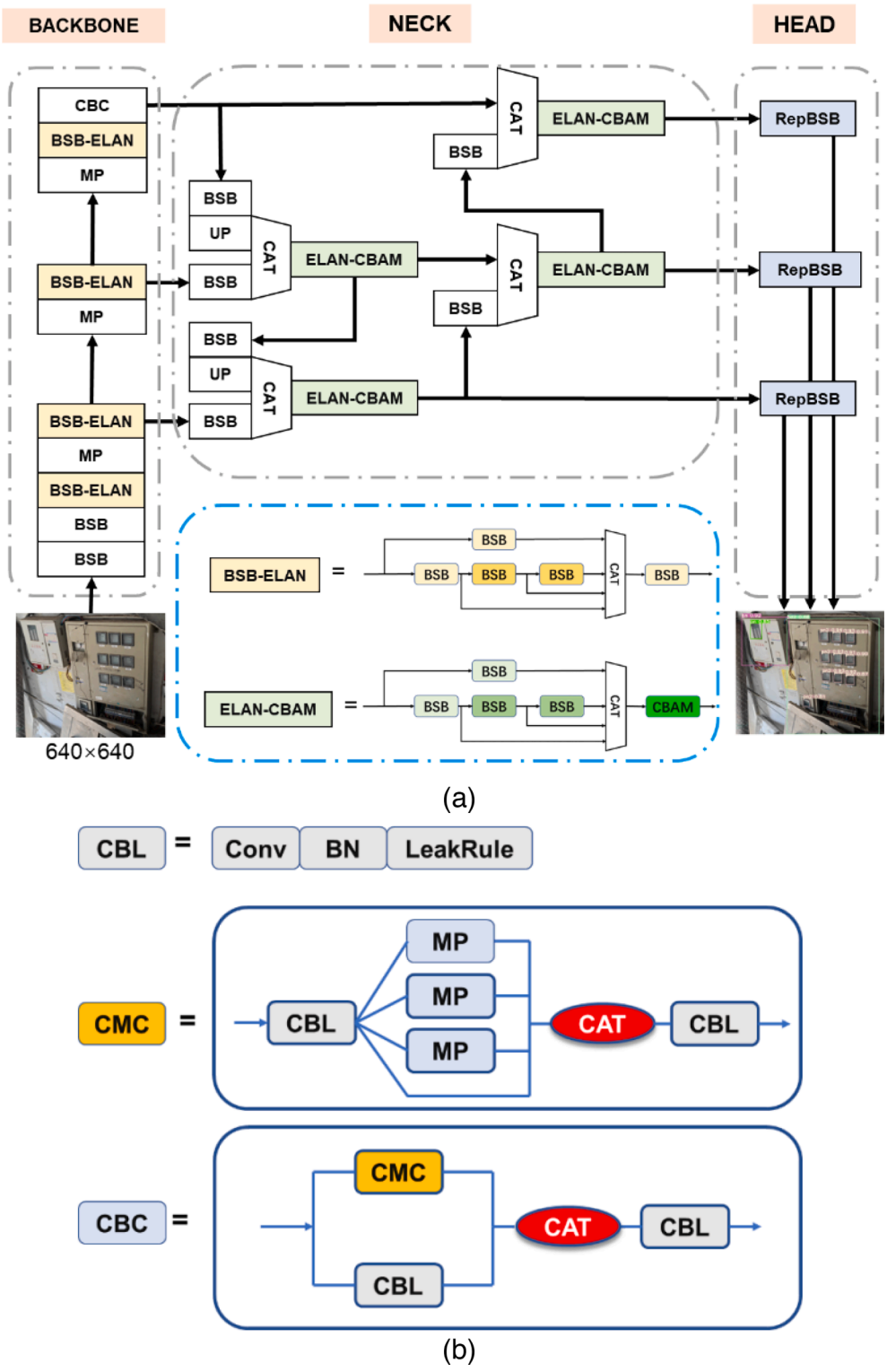


Fig. 2. The framework of the proposed EMB-YOLO Model. "BSB" is our improved convolutional module, and specific details will be explained in the next section. "MP" represents the maxpool layer. "CBC" is shown in (b). "UP" represents the upsample layer. "CAT" represents the concatenation of two feature maps in the channel dimension.

2. Related work

2.1. Electric power inspection

Vision based power detection tasks are generally divided into power component identification and defect detection. Traditional detection methods tended to use external contour features, texture features, color features, or a combination of these features. Then power components detection can be achieved by Canny algorithm, Sobel algorithm, and Directional Gradient Histogram (HOG), etc. In reference (Aguilar et al., 2019), texture features of insulators were extracted from edge orientation information for insulator detection. In reference (Shuang et al.,

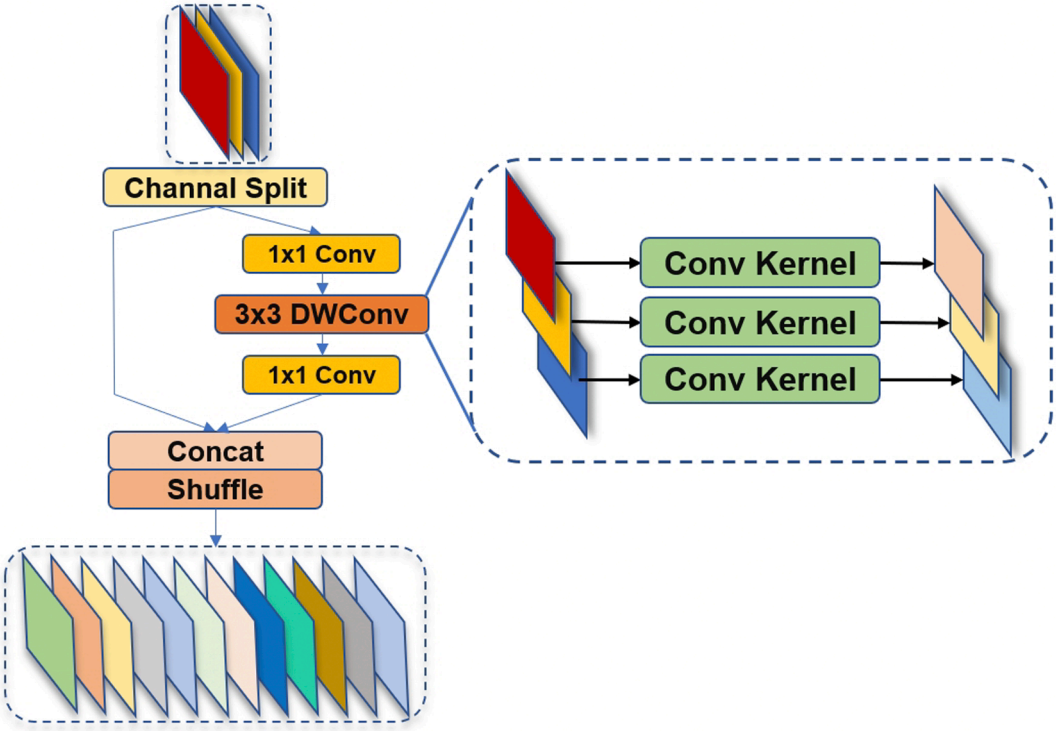


Fig. 3. ShuffleBlock schematic diagram.

2022), the image edges extracted by the Canny algorithm and then the criterion functions were constructed based on power line feature information, such as length, width, and direction, to get the exact power line regions.

In the environment of electric power inspection, there are often complex background environment, variable lighting conditions and small object in the image. Reference (Yang et al., 2022) proposes to embed an attention block in the power line detection network to solve the class imbalance problem. An attention fusion block is proposed for multi-scale feature fusion to obtain richer information and improve segmentation accuracy. In reference (Gao et al., 2021), a context branch combining with a spatial branch realizes the segmentation of power lines. The former generates useful global information and the latter preserves high-resolution segmentation details. In terms of defect detection, references (Tao et al., 2020)achieved the localization of insulator defects based on YOLO. However, during the inspection process, detecting the background of the object was often complex and easily obscured. In response to this problem, reference (Shuai et al., 2021)fused the convolutional block attention model on the basis of YOLOv3 and proposed a YOLOv3 transmission line fault detection method based on the convolutional block attention model; thereby improving the saliency of the fault object region in the image. In addition, Gaussian function was used to improve the non-maximum suppression and Focal Loss was introduced into the loss function to improve the detection accuracy of the occluded part of the object. In response to the problem of multi-scale features of the object detected in complex patrol environments, reference (Shuai et al., 2022) presents a YOLOv5 transmission line fault detection algorithm based on attention mechanism and cross scale feature fusion; which has constructed a multi-scale feature fusion module and a same scale feature weighted fusion module. The spatial channel attention module was introduced to enhance the saliency of the object region in complex backgrounds, and Bidirectional Feature Pyramid Network (BiFPN) (Tan and Efficientnet, 2019) was used as the model feature fusion method.

2.2. *Lightweight model*

Early image classification models established deeper networks to extract more features. From AlexNet (Krizhevsky et al., 2017) to Visual Geometry Group network (VGG) (Simonyan and Zisserman, 1409), then to GoogLeNet (Szegedy et al., 2015), Residual Network (ResNet) (He et al., 2016); and DenseNet (Huang et al., 2017), the model depth gradually increased and the model architecture gradually became complex. Although the model detection accuracy continued to improve, the number of model parameters continued to increase, making the model limited to use on devices with sufficient computing power. With the continuous development of Edge device such as mobile phones and UAVs in recent years, researchers began to focus on the lightweight of models. There are two main solutions available: one is to perform parameter pruning after training to reduce model parameters. The other is to design a more efficient convolutional calculation methods to reduce model parameters without compromising model performance. Typical examples of the first approach are Reparameterized Network (RepNet) (Wandt and Repnet, 2019) and RepLKNet (Ding et al., 2022). It uses a relatively large model in the training to extract sufficient features, and a small model based on large model reparameterization in the deployment to ensure sufficient detection speed. At present, the mainstream lightweight models include MobileNet (Howard et al., 2019); ShuffleNet (Ma et al., 2018); EfficientNet (Tan and Efficientnet, 2019); GhostNet (Han et al., 2020); etc. Many guiding methods have been proposed in lightweight models such as MobileNet and shuffleNet. For example, in MobileNet, a combination of deep convolution and point convolution was used, which could make the parameter quantity only 1/9 of that of ordinary convolution. In order to reduce the computational complexity of pointwise convolution in MobileNet, a strategy combining group convolution and channel shuffling was utilized in ShuffleNet. The above two lightweight techniques and ideas have also been used to improve current object detection models, such as the extensive use of RepConv in YOLOv6 to construct model structures. For real-time detection tasks, the YOLO series have considerable advantages. YOLOv5-s, YOLOv6-n and YOLOv7-tiny versions have been launched in YOLOv5, YOLOv6 (Li

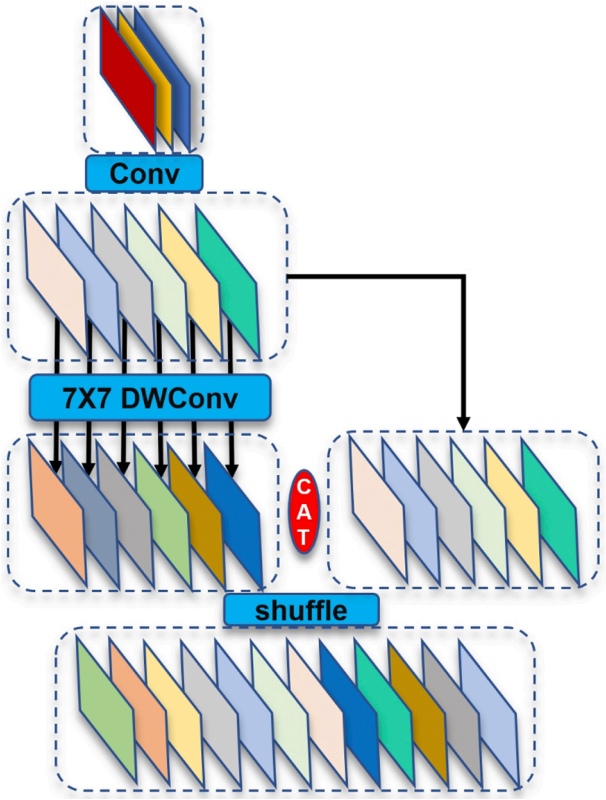


Fig. 4. Schematic diagram of Big Kernel ShuffleBlock. Different colors represent feature maps of different channels. "CAT" represents the concatenation of two feature maps in the channel dimension. "7 × 7 DWConv" represents a deep separable convolution with a convolution kernel size of 7.

et al., 2209) and YOLOv7 (Wang et al., 2023) to meet the detection requirements of Edge device. Nevertheless, the lightweight iterations of these algorithms, driven by the pursuit of swifter detection speeds, have adopted strategies involving the uncomplicated removal of specific feature extraction layers, exemplified by YOLOv5-s, or the streamlining of the model architecture, as observed in YOLOv7-tiny.

2.3. Attention mechanism

Attention mechanism has been widely applied in deep learning models. The channel-attention mechanism in Squeeze-and-Excitation Network (SE-Net) (Hu et al., 2018) enhanced the channel weight of feature maps that were useful for the current task, and suppressed feature channels that were not useful for the current task. In order to avoid dimension reduction in Squeeze-and-excitation attention mechanism and capture cross channel interaction information more effectively, Efficient Channel Attention Network (ECANet) (Wang et al., 2020)used ECA attention mechanism module and one-dimensional convolution to effectively capture cross channel interaction. The above two methods only utilized image channel information but ignored spatial information. The CBAM (Woo et al., 2018) attention mechanism focused on both channel domain analysis and spatial domain analysis; proposing a sequential attention structure from channel to space. Spatial attention focused on the pixel regions that played a decisive role in classification in the image while ignoring irrelevant areas. Channel attention processed the allocation relationship of feature map channels, and simultaneously allocated attention to two dimensions to enhance the performance improvement effect of attention mechanism on the model.

3. The proposed system

In this section, we will first introduce our model structure as a whole. Then, we will explain our improvement points and the ideas in sequence.

3.1. Network architecture

The schematic representation of EMB-YOLO is shown in Fig. 2. EMB-YOLO is an enhancement of YOLOv7-tiny, encompassing improvements in the backbone, neck, and head. The input image pixels of the backbone are 640 × 640. The backbone adopts BSB module which has increased the effective receptive field of the model while reducing the number of parameters, ensuring sufficient feature extraction. The backbone also included three maxpooling layers to obtain feature maps of different sizes. The neck of EMB-YOLO is constructed by path aggregation network (PAN), utilizing feature maps of different sizes obtained from the backbone model. After top-down feature fusion, bottom-up feature fusion is performed, making feature fusion more comprehensive and reducing the loss of shallow information. The use of ELAN-CBAM in feature fusion enables the model to focus on more valuable features to reduce noise interference. In the detection head, RepBSB is constructed from the idea of reparameterization to improve the performance of the Edge device end. The detailed introduction of each module is as follows.

3.2. Big kernel ShuffleBlock (BSB)

ShuffleNet v2 uses ShuffleBlock to build a model, as shown in Fig. 3.

In shuffleblock, the feature map undergoes an initial channel segmentation, resulting in two distinct groups. These two sets of feature maps traverse through two branches: one is a shortcut branch facilitating feature reuse, while the other uses a linear bottleneck structure. Within the linear bottleneck, the middle part adopts deep separable convolution (DWconv) which can reduce parameters. Preceding and succeeding the DWconv, two 1 × 1 convolution kernels are employed to fulfill the roles of dimension growth, dimension reduction and channel information integration.

A linear bottleneck structure can be expressed as follows:

$$w^{(3)}(x) = C^{(1)}\left\{D^{(3)}\left[C^{(1)}(s_1(x))\right]\right\} \tag{1}$$

where, $w^{(k)}(x)$ represents a linear bottleneck structure with a deeply separable convolution kernel size of k, $C^{(k)}(x)$ represents a normal convolution with a convolution kernel size of k, $s_i(x)$ represents the i th branch after channel segmentation, and $D^{(k)}(x)$ represents a deeply separable convolution with a convolution kernel size of k.

Finally, the two branches are spliced together again, and then the order of the channels is randomly disrupted to realize the information exchange between the channels, that is,

$$sh\left[w^{(3)}(x) \oplus s_2(x)\right] \tag{2}$$

where $sh(x)$ represents shuffle operation and $\oplus$ represents concat operation.

Although this linear bottleneck structure widens the model and extracts more features, a large number of 1 × 1 convolutions consume hardware resources and increase the cost of memory access. Moreover, the channel information integration function of 1 × 1 convolution is unnecessary under the channel shuffle operation, so we abandon the 1 × 1 point-by-point convolution and only use the deep separable convolution. In addition, to ensure the communication of information between channels and the transmission of information in the original feature map, the shortcut branch is retained, and the schematic diagram of the module is shown in Fig. 4.

Our proposed convolution module performs full channel dense convolution on the feature map at first. Then input the obtained feature

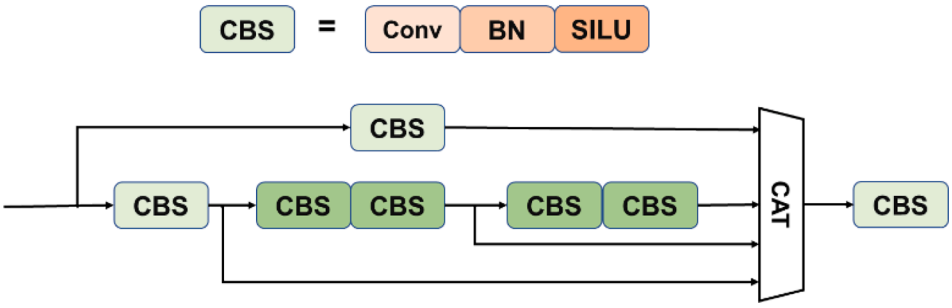


Fig. 5. Detailed diagram of ELAN module.

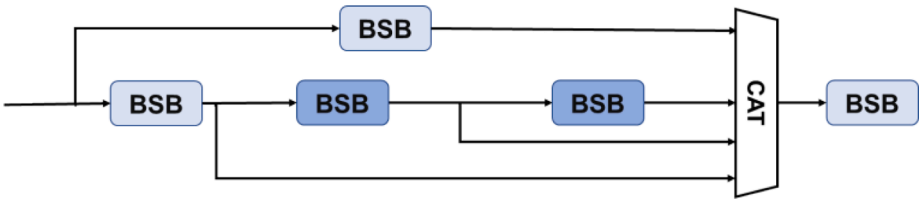


Fig. 6. Detailed diagram of BSB-ELAN module.

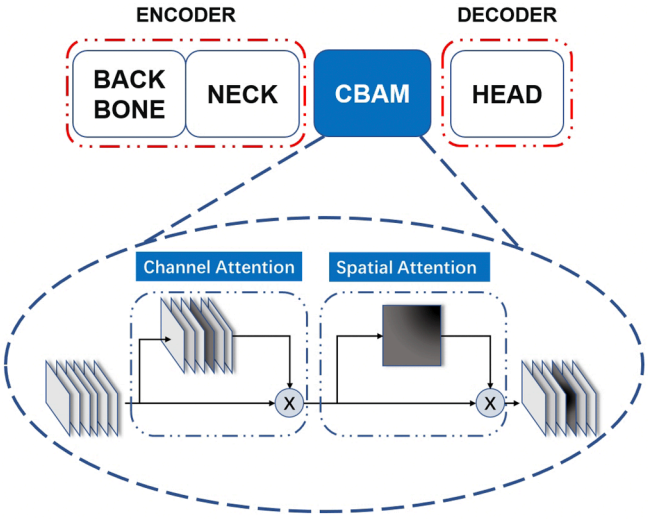


Fig. 7. Simplified model structure diagram with CBAM added.

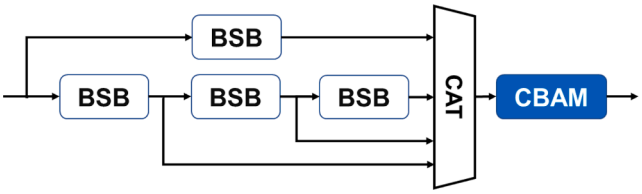


Fig. 8. ELAN-CBAM schematic diagram.

maps into the shortcut branch and the depth separable convolution branch respectively. After that, concatenate the feature maps obtained from the two branches. Finally, the concatenated feature map is randomly shuffled in the channel dimension to obtain the output feature map. Similarly, BSB modules can be represented as follows:

$$B^{(3)}(x) = sh\{D^{(7)}[f(x)] \oplus f(x)\} \tag{3}$$

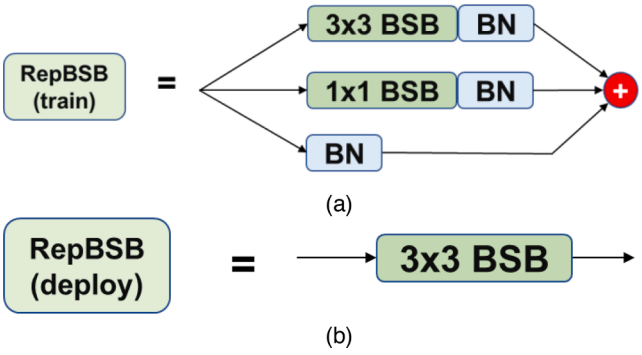


Fig. 9. RepBSB schematic diagram. '+' indicates the addition of feature maps.

3.3. BSB-ELAN and ELAN-CBAM

In YOLOv7, ELAN is used to control the shortest and longest gradient path, which enable a deeper model to learn and converge more effectively. The structure is shown in Fig. 5.

But the effective receptive field of stacking multiple small convolutional kernels is not as good as that of a single large convolutional kernel. Therefore, we have retained the basic framework of ELAN and replaced two consecutive 3 × 3 convolutional kernels with a 5 × 5 BSB module, as shown in Fig. 6.

BSB-ELAN module can be represented as follows:

$$BSB-ELAN(x) = B\left\{\begin{array}{c} B^{(3)}(x) \oplus B^{(3)}(x) \oplus B^{(5)}\left(B^{(3)}(x)\right) \\ \oplus B^{(5)}\left(B^{(5)}\left(B^{(3)}(x)\right)\right) \end{array}\right\} \tag{4}$$

In addition, as shown in Fig. 7, we add the CBAM attention mechanism after the neck part of the model. The visual attention mechanism enables the model to focus on the region of interest, which is conducive to identifying and locating defects in the meter box under various complex backgrounds.

Our approach to joining is to embed CBAM into the BSB-ELAN module, forming ELAN-CBAM, as shown in Fig. 8.

The ELAN-CBAM module can be represented as follows:

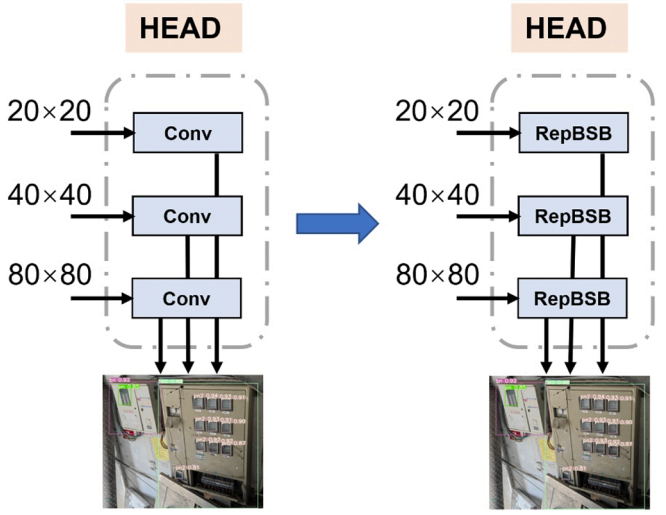


Fig. 10. Improvement schematic diagram of EMB-YOLO detection head.

$$ELAN - CBAM(x) = CBAM \left\{ \begin{matrix} B^{(3)}(x) \oplus B^{(3)}(x) \oplus B^{(5)}\left(B^{(3)}(x)\right) \\ \oplus B^{(5)}\left(B^{(5)}\left(B^{(3)}(x)\right)\right) \end{matrix} \right\} \tag{5}$$

3.4. RepBSB

In the practice of power inspection, the computational resources after model deployment are scarce. Therefore, in response to this situation, we use the idea of structural reparameterization to construct the detection head of EMB-YOLO. We propose a RepBSB structure based on the idea of reparameterization. The structure of RepBSB is divided into two stages. During the model training phase, RepBSB(train) as shown in Fig. 9(a). During the model deployment phase, RepBSB(deploy) as shown in Fig. 9(b).

The RepBSB module can be represented as follows:

$$RepBSB_{train}(x) = \sum_{k=0,1,3} BN\left[BSB^{(k)}(x)\right] \tag{6}$$

Definition $BSB^{(0)}(x) = x$.

$$RepBSB_{(deploy)} = B^{(3)}(x) \tag{7}$$

Using the proposed RepBSB, we can improve the head part of the model, as shown in Fig. 10.

4. Mathematical analysis

4.1. Parameter quantity analysis

In convolution, parameter quantities calculation can be expressed as:

$$k \times k \times C_i \times C_o \tag{8}$$

where k is the size of the convolution kernel, C_i is the number of input feature map channels, C_0 is the number of output feature map channels.

According to formula (8), the common 3×3 convolution parameter used in YOLOv7 is:

$$3 \times 3 \times C_i \times C_o \tag{9}$$

But the parameter quantity of BSB module can be expressed as:

$$\frac{3 \times 3 \times C_i \times C_o}{2} + \frac{7 \times 7 \times 1 \times C_o}{2} \tag{10}$$

After subtracting formula (10) from formula (9), we can get that the reduction of the module parameters proposed by us compared with the ordinary 3×3 convolution parameters is:

$$\frac{9C_i - 49}{2} C_o \tag{11}$$

As long as $C_i > 5$, formula (11) will be positive, meaning that the parameters of BSB will be less than those of common 3×3 convolution. And the gap will continue to increase as C_i becomes larger. It can be seen from the above analysis that although the proposed convolution module uses 7×7 big kernel convolution, the reasonable use does not lead to the increase of convolution parameters, on the contrary, it decreases.

4.2. Receptive field analysis

Due to the dense connections of neurons in neural networks, a big convolution kernel means a larger number of parameters and calculations. But the stacking of multiple 3×3 convolution kernels on the theoretical receptive field can completely replace the big convolution kernel, and there will be less parameters. However, if we analyze the effective receptive field, the results would be different. In fact, within the receptive field, the closer the unit to the center, the greater the impact on the unit, so the effective receptive field deserves attention. In addition, the object detection task needs to locate the object, and the effective receptive field is more significant for the downstream tasks of the classification model.

We will explain the mathematical notation that will appear at first. We use $w(m)$ to express the weight of the m-th pixel of the convolution kernel, and $o(t)$ to represent the gradient signal of the input pixel. If $w(m)$ is normalized, then $o(t)$ is exactly equal to the probability $p(S_n = t)$, where $S_n = \sum_{i=1}^{n} X_i$. X_i is a polynomial variable that is independent and identically distributed with $w(m)$, that is, $p(X_i = m) = w(m)$. Therefore, according to the central limit theorem, when $n \to \infty$, the distribution of $\sqrt{n}\left[\frac{1}{n}S_n - E(X)\right]$ converges to Gaussian distribution $N(0, var[X])$. This means that when n is large enough, S_n will be a Gaussian distribution with the mean $E(X)$ and the variance $nvar[X]$, and then $o(t)$ also satisfies the Gaussian distribution. The mean and variance of S_n are calculated as follows:

$$E[S_n] = n \sum_{m=0}^{k-1} m w(m) \tag{12}$$

$$Var[S_n] = n\left(\sum_{m=0}^{k-1} m w(m) - \left(\sum_{m=0}^{k-1} m w(m)\right)\right)^2 \tag{13}$$

This indicates that $o(t)$ decays exponentially from the center of the receptive field. The decay rate is related to the variance of Gaussian distribution. The radius of the effective receptive field can be approximately equal to the standard deviation, that is,

$$\sqrt{Var[S_n]} = \sqrt{nVar[X_i]} = O\left(\sqrt{n}\right) \tag{14}$$

Because the theoretical receptive field increases linearly with the deepening of the model depth, the actual contraction rate of the effective receptive field is $O(1/\sqrt{n})$。.

If $w(m) = 1/k$, then:

$$Var[S_n] = n\sqrt{\sum_{m=0}^{k-1} \frac{m^2}{k} - \sum_{m=0}^{k-1} \frac{m}{k}} = \sqrt{\frac{n\left(k^2 - 1\right)}{12}} = O\left(k\sqrt{n}\right) \tag{15}$$

Therefore, in the simple case of uniform weighting, the effective receptive field size increases linearly with the convolution kernel size k. To sum up, the size of the effective receptive field is proportional to the size of the convolution kernel and the square root of the depth of the model. Therefore, it is actually more efficient to achieve a large receptive field by stacking layers than to increase the size of the convolution kernel. Although the model employed multiple 3×3 convolutional kernels to increase the receptive field range, the effective receptive field

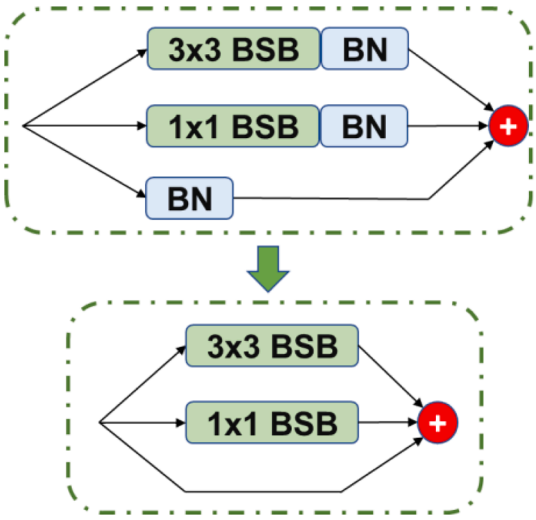


Fig. 11. The fusion of convolutional layer and BN layer.

is not as good as the shallow big convolutional kernel model. In Section V, Part C, we visualized the effective receptive field to verify our analysis.

4.3. Reparameterization analysis

There is a BN layer formula:

$$BN(x) = \gamma \frac{(x - mean)}{\sqrt{var}} + \beta \tag{16}$$

Substituting formula (6) into formula (16) yields:

$$RepBSB_{train}(x) = \sum_{k=0,1,3} \gamma_k \frac{\left[BSB^{(k)}(x) - mean_k\right]}{\sqrt{var_k}} + \beta_k$$

$$= \frac{\gamma_k}{\sqrt{var_k}} \sum_{k=0,1,3} BSB^{(k)}(x) + \sum_{k=0,1,3} \frac{-\gamma_k mean_k}{\sqrt{var_k}} + \beta_k \tag{17}$$

Obviously, the convolutional layer plus BN layer is essentially a convolutional layer with BN layer parameters and bias. It can be represented by the following formula.

$$BSB^{(k)}_{fuse}(x) = \frac{\gamma_k}{\sqrt{var_k}} \sum_{k=0,1,3} BSB^{(k)}(x) \tag{18}$$

$$BIAS^{(K)}_{fuse} = \sum_{k=0,1,3} \frac{-\gamma_k mean_k}{\sqrt{var_k}} + \beta_k \tag{19}$$

Therefore, we can fuse the convolutional layer with BN layer structure into a convolutional layer, and the fusion result can be expressed as:

$$RepBSB_{fuse}(x) = BSB^{(k)}_{fuse}(x) + BIAS^{(K)}_{fuse} \tag{20}$$

During parameter fusion, the fusion of convolutional layer and BN layer is shown in the Fig. 11.

In convolutional fusion, identity can be equivalent to a 1×1 convolutional kernel with a weight value of 1. For a 1×1 convolution, it can be expanded to a 3×3 convolution by filling in 0 with excess parts. By accumulating three 3×3 convolutions, a new 3×3 convolution can be formed, completing the entire reparameterization operation and obtaining a simple and direct 3×3 convolutional kernel for final deployment as Fig. 9(b).

5. The experiment

In this section, we will briefly describe the constructed dataset, experimental environment details, and several model evaluation indicators. Then, we conduct ablation experiments to evaluate the performance of EMB-YOLO and compare it with other major advanced algorithms. In addition, visualization and some discussions are conducted.

5.1. Dataset

To train and validate the effectiveness of our proposed method, we construct a dataset (partial) as shown in Fig. 12, named EMB-11.

During data collection, we use camera embedded in the robot and mobile phone to capture images of indoor meter boxes in different indoor scenes. In addition, we also use UAV to take images of the meter box in different outdoor scenes with different weather conditions. Besides, 500 high-quality meter box images from various perspectives are collected from the Internet.

In the data cleaning stage, we remove some similar, incomplete, fuzzy and poor-quality images. This resulted in a refined dataset comprising 500 UAV view images, 1000 robot view images, 1000 mobile phone view images and 500 images collected from the Internet to build

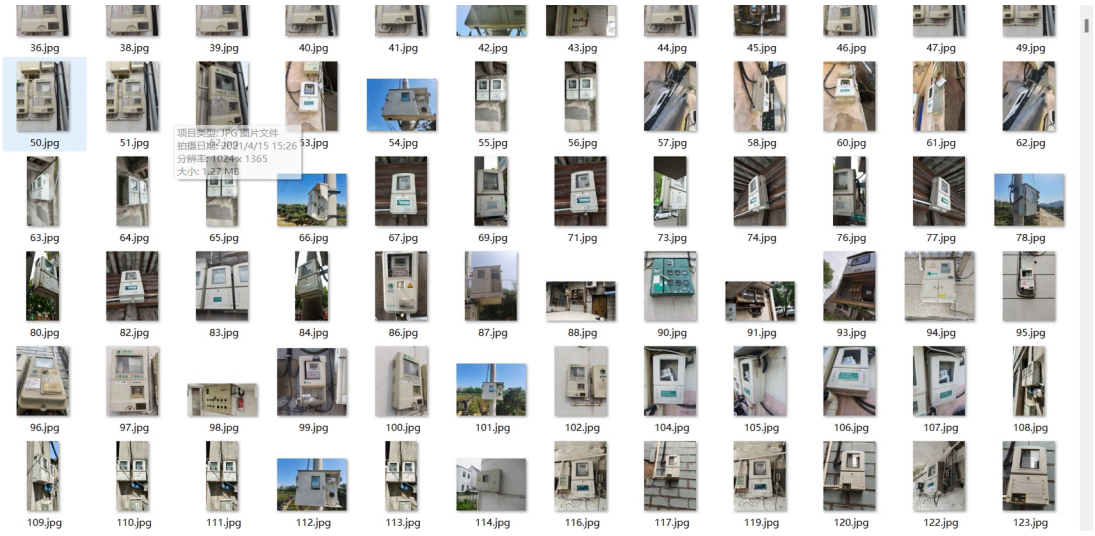


Fig. 12. Dataset of Electric Meter Box.

Fig. 13. Example of Electric Meter Box from Different Perspectives.

Table 1
Statistics of Various Labels in Dataset (For ease of display, the labels are abbreviated).

Class	Train	Val	Total	Proportion
Single box_a Normal	1790	430	2220	15.16 %
Multi box_a Normal	462	114	576	3.93 %
Single box_p Normal	2209	552	2761	18.85 %
Multi box_p Normal	2098	521	2619	17.88 %
Single box_a Damage	598	148	746	5.09 %
Multi box_a Damage	167	41	208	1.42 %
Single box_p Damage	284	73	357	2.44 %
Multi box_p Damage	476	143	619	4.23 %
Single box_seal Exist	916	272	1188	8.11 %
Single box_seal Miss	803	228	1031	7.04 %
Multi box_a corroded	240	73	313	2.14 %
Single box_complete	410	87	497	3.39 %
Single box_incomplete	401	107	508	3.47 %
Multi box_complete	378	113	491	3.35 %
Multi box_incomplete	421	93	514	3.51 %
Total	**11,653**	**2995**	**14,648**	\

the dataset. Examples of robot view, mobile phone view and UAV view are shown in Fig. 13 respectively.

During the data annotation stage, to ensure the quality of the EMB-11 dataset, we formulate strict annotation standards and used three professional image annotation workers. The used labeling software is LabelImg. Each object appearing in the image is individually labeled, avoiding the annotation of adjacent objects with a single annotation box. Additionally, due to the shooting angle, over one-third of the objects that do not appear in the image are not labeled. In the first round of annotation, three professional annotators manually label the category and location of the objects. In the subsequent round of annotation, three annotators review the labels and collectively determine whether every annotation adherence to the established standards.

It is noteworthy that in practical scenarios, normal meter boxes are detected far more frequently than damaged ones, and certain types of meter boxes outnumber others. This introduces potential data bias between positive and negative samples. To avoid this situation, we filtered out many poor-quality positive sample labels and collected as many negative sample labels as possible, balancing the number of positive and negative samples in label processing.

Finally, the EMB-11 dataset contains a total of 3000 images and 14,648 labels, with resolutions ranging from 152 × 202 to 3384 × 6016 pixels. The labels are annotated in Pascal VOC format, encompassing six defect types and 15 distinct labels. The specific types of labels can be seen in Fig. 16. The detail of labels is shown in Table 1.

Examples of six types of defects are shown in the Fig. 14.

5.2. Evaluation metrics and training process

To verify the effectiveness of the proposed method, we use two widely used evaluation indicators, i.e., mAP_{50} and $mAP_{50:95}$, to quantitatively evaluate the performance of defect detection methods (Everingham et al., 2010). mAP_{50} is the area that under the Precision-Recall (P-R) curve when the IOU threshold is set to 0.5. It signifies the average accuracy of all categories, which can characterize the feature extraction ability of the model as a whole. $mAP_{50:95}$ refers to the calculation of mAP every 0.05 when the IOU is from 0.5 to 0.95, and the average value is taken. This not only characterizes the overall feature extraction ability of model but also demonstrates the positioning ability during the detection process.

In addition, to consider the real-time requirements after the model is deployed on the Edge device, we use the model parameter quantity (Mb) to represent the learnable parameter quantity in the model, and FPS to represent the number of pictures that can be detected per second.

During the training process, the total number of images is 3000, with 80 % being used as the training set (2400 images) and 20 % as the validation set (600 images). All images will be scaled to 640 × 640. During the training phase, we use the SGD optimizer with the momentum parameter set to 0.937 and the initial learning rate set to 0.01. The total number of epochs is 600, and the batch size is 64. All experimental results are obtained from the Python platform on NVIDIA GPU, with the following server configuration: CPU model Intel Xeon Gold twelve core 6136 @ 3.00 GHz, GPU model NVIDIA TITAN V 12 GB × 8 @ 1455 MHz, operating system Ubuntu 16.04, CUDA version 10.2, and Python version 1.8.0.

The loss curve and Precision-Recall curve during the training process is shown in Fig. 15 and Fig. 16 respectively.

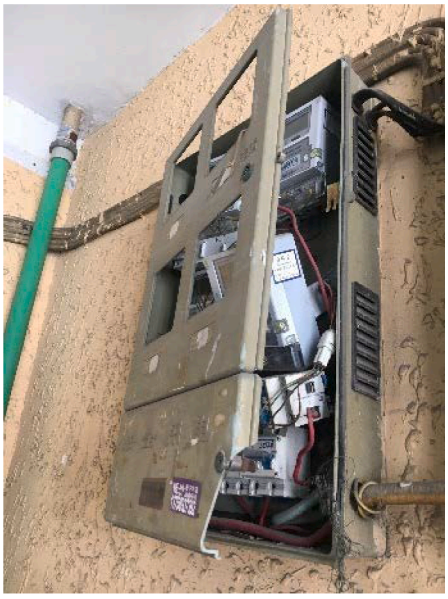
Fig. 15 reflects the change in the value of the loss function with the number of training steps. During the training process, we can observe the loss curve to determine whether the model converges or not. From the curve, it can be seen that in the later stage of training, both the training loss curve and the validation loss curve have approached a plateau, and the difference between them is relatively small, indicating that the training has reached a relatively satisfactory state. Fig. 16 shows the prediction performance of the model after training more accurately.

5.3. Ablation experiments

In order to investigate the impact and effectiveness of BSB, BSB-ELAN, ELAN-CBAM, and RepBSB in our model, we conduct ablation experiments using YOLOv7-tiny as the reference model and discuss them. The experimental results on EMB-11 are shown in TABLE II.

Firstly, considering that YOLOv7 doubled the number of channels in the original feature map across all three detection heads, we examined the necessity of widening the model width in the detection head. Unlike YOLOv7, we opted to maintain the same number of channels as the last layer of the detection neck. From the experimental results, it can be seen that although using detection head with more channels can indeed bring better prediction accuracy, but parameters and computational overhead cannot be ignored. The adjustment to reduce the number of channels in the detection head resulted in a commendable 13.1 % reduction in model parameters, with only a marginal decline of 0.4 % in mAP_{50} and 0.1 % in $mAP_{50:95}$. Considering the imperative balance between speed and accuracy, we deem this trade-off worthwhile. Subsequent

(a) Single box_a Damage



(b) Multi box_a Damage

(c) Single box_p Damage

(d) Multi box_p Damage

(e) Single box_seal Miss

(f) Multi box_a corroded

Fig. 14. Example Diagram of Six Types of Defects.

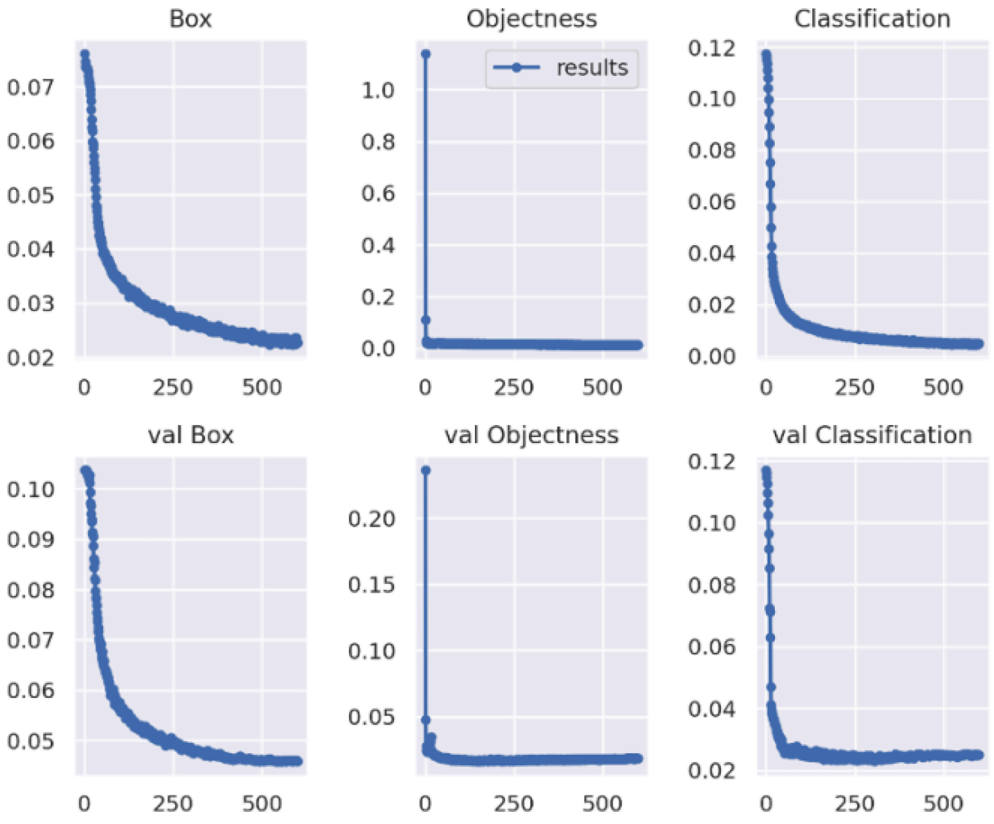


Fig. 15. Training loss and validation loss curve. The abscissa represents the epoch count, and the ordinate represents the loss value. The plot on the first row depicts the loss curve during training, while the second row represents the loss curve during validation.

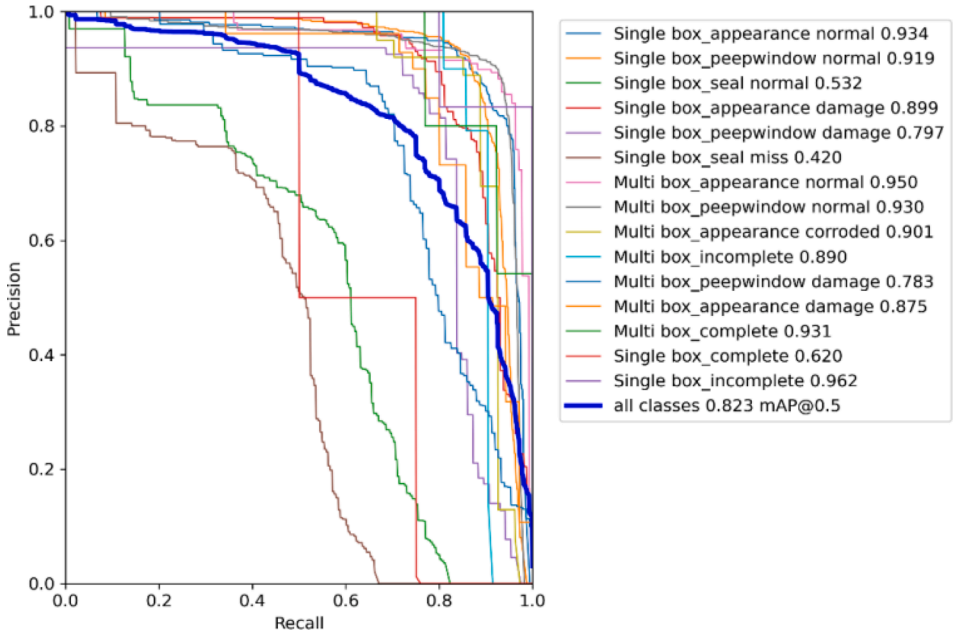


Fig. 16. Precision-Recall curve.

experiments will consistently adopt the channel reduction operation (marked as ※ in the Table 2).

Next, we replace all convolutional layers in YOLOv7-tiny with BSB module. It can be seen that after replacing the original convolutional layer with the BSB module, the parameter count and computational cost of the entire model have significantly decreased. But at the same time, the model becomes smaller, the ability of feature extraction is weakened. Compared to the original model, it is inevitable for mAP_{50} to decrease by 2.5 %. However, the reduction of parameters and computational overhead (FLOPS) are 54.9 % and 50.4 % respectively. This shows that the BSB module is very effective in reducing the parameters of the model, and will not lead to a significant reduction in the accuracy of the model. Then we will try to use some other techniques to recover the 2.5 % accuracy loss without significantly increasing the parameters

Table 2
Results of Ablation Experiment.

※	BSB	BSB-ELAN	C	R	N	mAP50 (%)	$mAP50_{:95}$ (%)	Params (Mb)	FLOPS(G)
					3	79.3	65.7	6.1	13.3
√					3	78.9	65.6	5.3	11.8
√	√				3	76.8	65.3	**2.7**	**6.6**
√	√	√			5	80.4	69.0	4.1	10.9
√	√	√		√	5	80.5	69.0	4.6	11.8
√	√	√	√		5	80.6	69.7	4.3	11.1
√	√	√	√	√	5	81.2	70.2	4.7	12.0
√	√	√	√	√	7	**82.8**	**71.2**	4.8	12.4
√	√	√	√	√	9	80.9	69.4	4.9	13.0
√	√	√	√	√	11	80.3	69.8	5.0	13.7

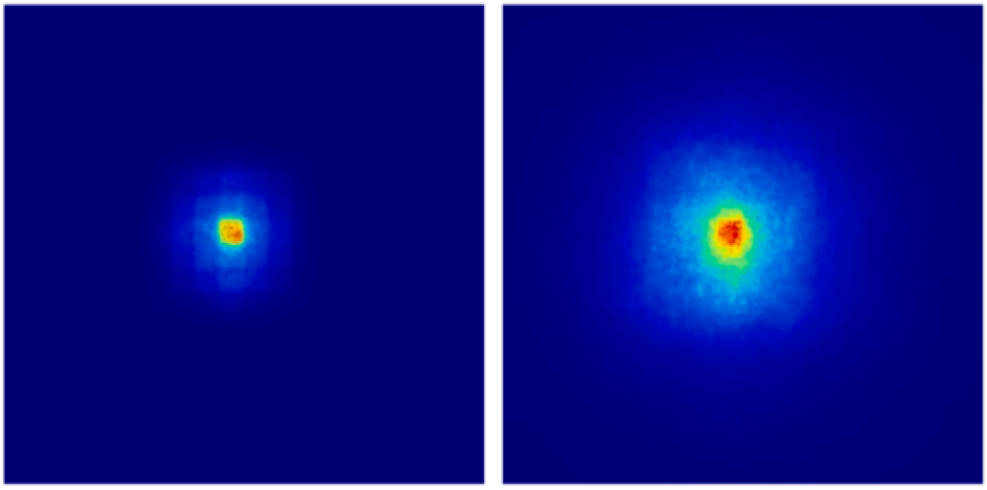


Fig. 17. 3 × 3 and 5 × 5 Comparison of convolutional kernel effective Receptive field.

Table 3
Results of Different CBAM Module Addition Methods.

Method	mAP_{50} (%)	$mAP50_{:95}$ (%)	Params (M)	FLOPS (G)
MIX CBAM	**81.2**	**70.2**	**4.72**	12.0
ADD CBAM	80.6	69.1	4.73	12.0

of the model.

Subsequently, we replace the ELAN structure with BSB-ELAN. Before and after using BSB-ELAN, the effective receptive field of the last layer of the backbone part of the model is shown in Fig. 17. Obviously, big convolutional kernel brings more effective receptive field, which enables the model to obtain more global information. Therefore, the experimental result show that $mAP_{50:95}$ is increased by 3.3 % compared with the benchmark model, while mAP_{50} is increased by 1.1 %. At the same time, compared with the benchmark model, the parameters are still reduced by 32.8 %. This shows that our rational use of big convolution kernel does not bring about a huge increase in parameters, but can significantly improve the positioning ability of the model to the object.

Then, we add ELAN-CBAM to the model. We discuss the combination method of ELAN and CBAM, and the experimental results are shown in Table 3.

MIX CBAM refers to replacing the last convolutional layer of ELAN with CBAM directly, while ADD CBAM refers to adding a separate layer of CBAM to ELAN. It can be seen that directly replacing the last convolutional layer with CBAM will have better results, with smaller parameter count and computational overhead. Therefore, EMB-YOLO adopts the MIX CBAM approach. The heatmap of the output layer before and after the addition of ELAN-CBAM is shown in Fig. 18. Obviously, ELAN-CBAM can focus on more valuable features in complex backgrounds, provide more accurate positioning information, reduce background noise interference, and increase the robustness of the model.

Finally, we use RepBSB in the detection head. We firstly evaluate the performance of RepBSB and ELAN-CBAM when acting alone, and then compare the effect when using different sizes of convolution kernels in BSB. From the experimental results in the third part of TABLE II, it can be seen that both RepBSB and CBAM have varying degrees of improvement when they act alone. But when they act together, they are better than when they act alone. At the same time, they do not bring more parameter quantities and computational costs. In addition, it can be seen that when the convolution kernel size of the group convolution in BSB is set to 7, the detection performance of the model is the best. The mAP_{50} has increased by 3.5 %, especially on the index $mAP_{50:95}$ which can reflect the positioning ability, it has increased by 5.5 % compared with the benchmark model YOLOv7-tiny. This result shows that our model has better recognition ability and positioning ability. However, experiment also reveals that if the size of the convolutional kernel is further increased, it not only leads to an increase in parameters but also reduces the accuracy of detection. This phenomenon may be due to the oversized convolutional kernel overlooking certain local information. As well known, each convolutional kernel is responsible for capturing image information, which might be edge contour information or target information. Large convolutional kernels generally trade off more parameters and computation for capturing high-resolution information with more details. However, as the size of the convolutional kernel increases, and the kernel size nearly equals the input resolution, the convolutional network approaches a fully connected network. Such a fully connected design approach is inferior in accuracy and performance to

Fig. 18. Comparison of heatmaps before and after adding ELAN-CBAM.

Table 4
Results of Different Algorithms on EMB-11.

Method	mAP_{50}(%)	$mAP50_{:95}$ (%)	Params (Mb)	FLOPS (G)	FPS
SSD	70.6	51.0	24.9	277.6	35
Faster R-CNN	78.1	63.4	136.9	401.9	5
YOLOX-tiny	75.9	62.9	5.04	15.2	256
YOLOv5-s	79.1	66.0	7.23	16.5	208
YOLOv6-n	78.6	67.2	**4.30**	**11.1**	**357**
YOLOv7-tiny	79.3	65.7	6.05	13.3	294
OUR	**82.8**	**71.2**	4.82	12.4	343

convolutional neural networks. Therefore, oversized convolutional kernel size may compromise accuracy and performance. To enhance model accuracy and performance, an appropriate convolutional kernel size should be selected to capture both high-resolution and low-resolution information as much as possible.

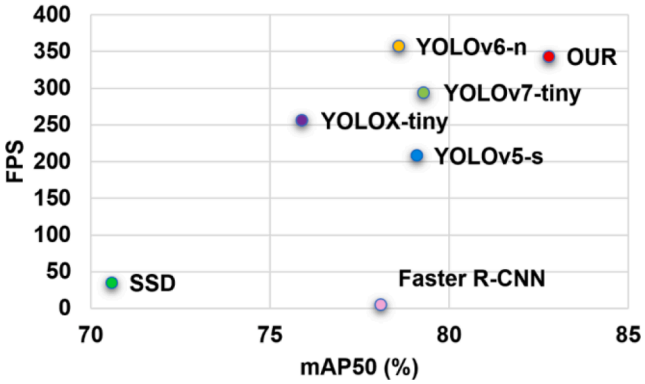


Fig. 19. Comprehensive Comparison of Algorithm Performance.

5.4. Comparison experiments

5.4.1. Quantitative experiment

In order to objectively evaluate EMB-YOLO, we conduct a

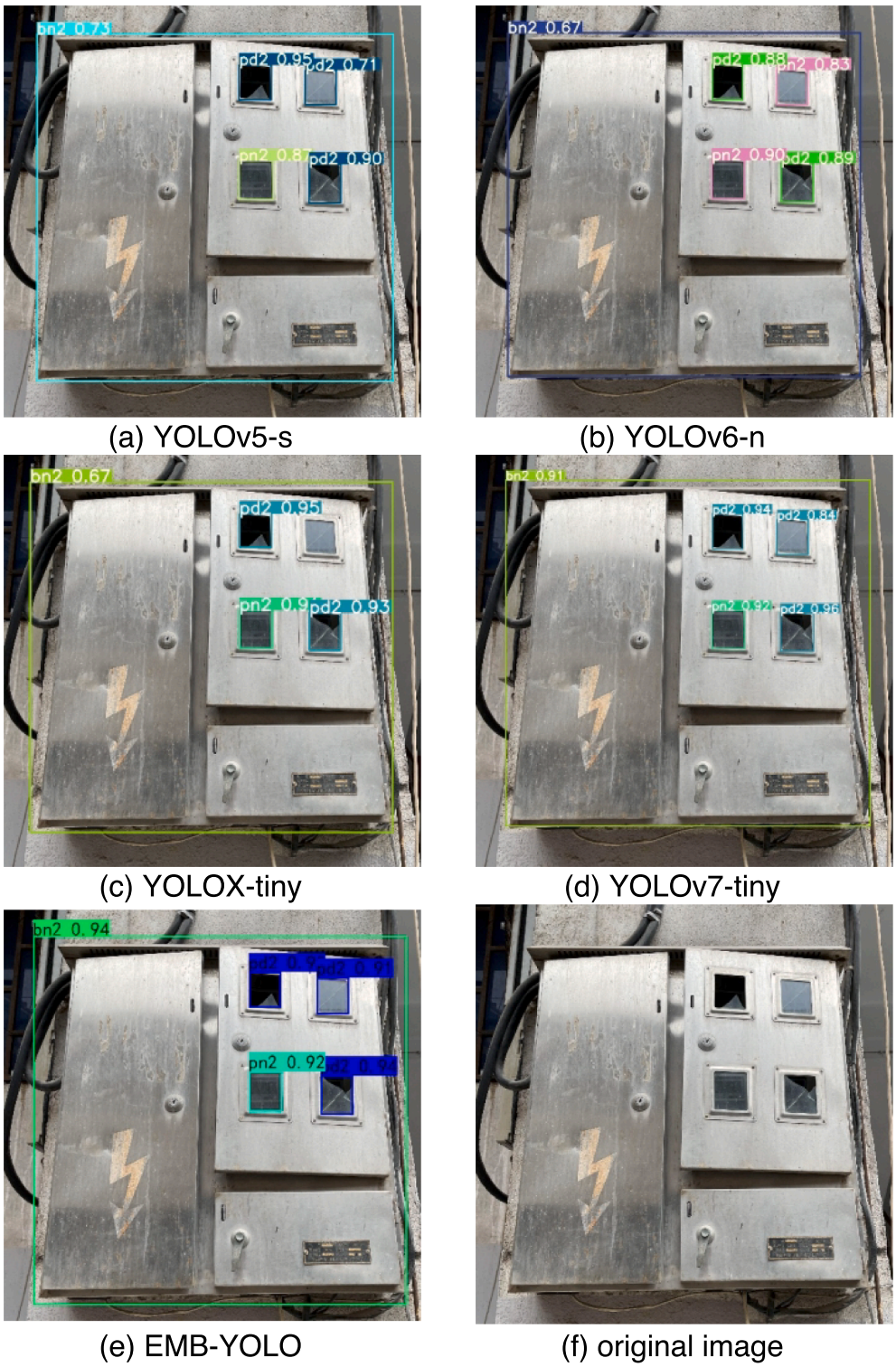


Fig. 20. Example 1 of Meter Box Detection.

longitudinal comparison with current mainstream object detection algorithms and establish a benchmark on our dataset. The YOLO series of algorithms have been iterated in many versions and are widely applied in industrial practice. Therefore, we mainly choose YOLO series algorithms for comparison. In order to reflect the progressiveness of our algorithm, we have selected YOLOv5, YOLOv6, YOLOX, YOLOv7 released in recent three years. It is also worth mentioning that each of the above algorithms has several versions with different model sizes. For example, YOLOv5 has four versions: YOLOv5-s, YOLOv5-m, YOLOv5-l, and YOLOv5-x, to meet the different requirements of supporters for accuracy and speed. To ensure the rigor of the experiment, we selected YOLOv5-s, YOLOv6-n, YOLOX-tiny, and YOLOv7-tiny with similar model sizes as comparative algorithms. In addition to YOLO series, we also selected the classic one-stage algorithm SSD and the classic two-stage algorithm Faster R-CNN to show the progressiveness of our algorithm. The experimental environment and parameters are consistent, and FPS data is obtained on NVIDIA TITAN V. The experimental results are shown in the Table 4.

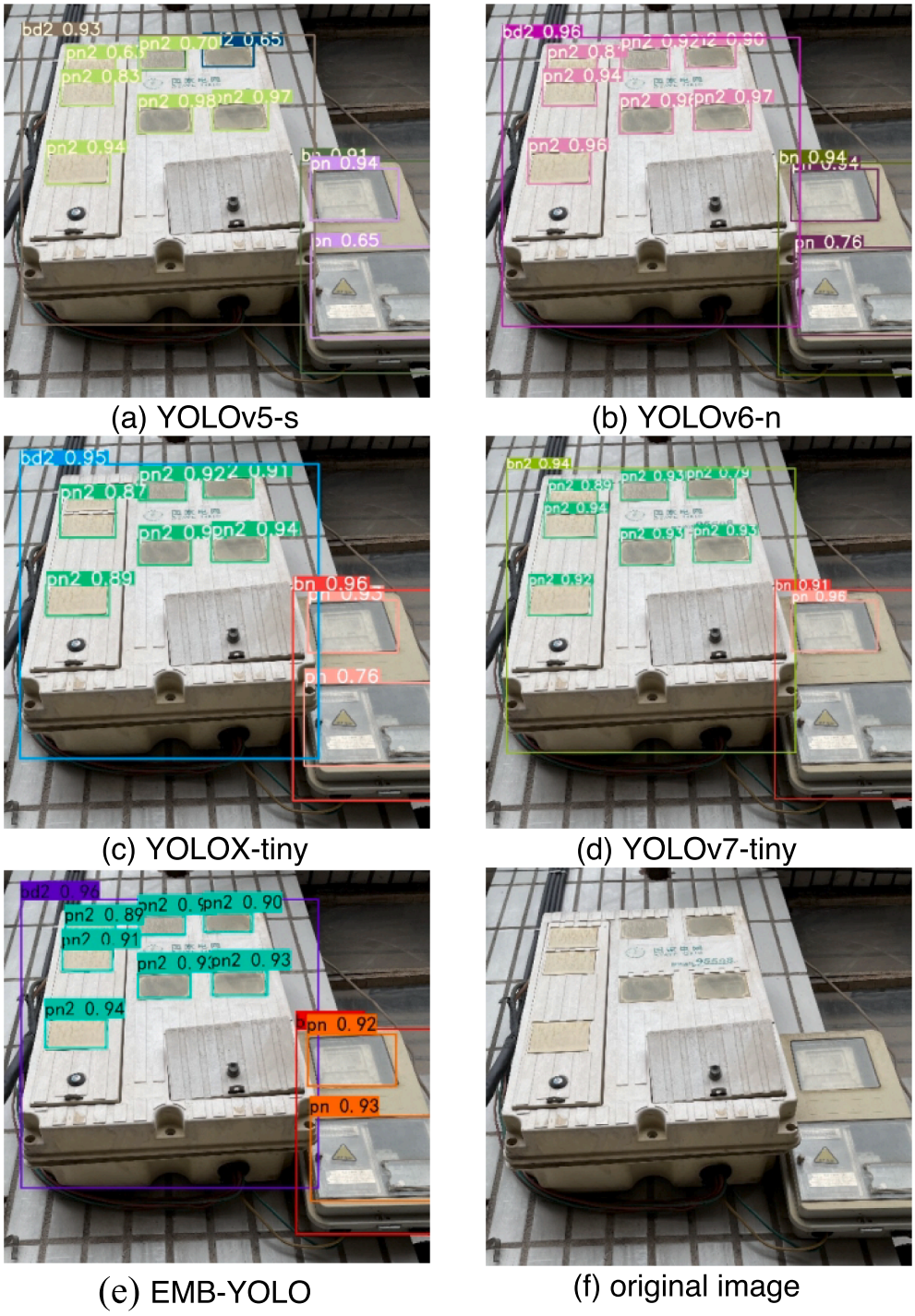


Fig. 21. Example 2 of Meter Box Detection.

Table 5
Results of Different Algorithms on RSIn-Dataset.

Method	mAP_{50}(%)	$mAP50_{:95}$(%)
SSD	85.0	44.5
Faster R-CNN	92.7	54.6
YOLOXt	92.3	54.2
YOLOv5s	93.1	54.9
YOLOv6n	92.6	54.2
YOLOv7t	92.5	54.6
Our	**93.5**	**55.2**

As can be seen in TABLE IV, our proposed algorithm has achieved advantages over the current advanced algorithms in the two indicators of mAP_{50} and $mAP_{50:95}$, which are respectively 3.5 % and 4 % higher than the other highest values. In addition, in terms of the comparison of model size and FPS, except for YOLOv6-n, which has a weak leading edge, the other algorithms lag behind our algorithm. However, although YOLOv6n has an advantage in the speed index, its mAP_{50} and $mAP_{50:95}$ is 4.2 % and 4 % lower than our algorithm respectively. Therefore, it can be said that our algorithm achieves the optimal balance between speed and accuracy in the compared algorithms.

Fig. 19 shows this more intuitively. The scatter diagram takes mAP_{50} as the horizontal axis and FPS as the vertical axis, which can

characterize the balance ability between detection speed and detection accuracy, and the higher the point on the upper right, the better its comprehensive performance. As can be seen from Fig. 19, our proposed method is at the top right, so it can achieve the best performance in general.

Although our method has reached the best accuracy on the constructed dataset, it failed to achieve the optimal speed index. After analysis, this phenomenon maybe caused by the following two reasons. Firstly, we not only used 3 × 3 convolution kernel, but also 5 × 5 and 7 × 7 convolution kernels have been employed. However, the traditional lightweight algorithms generally use 3 × 3 convolution kernel, and the existing computing library is also deeply optimized for 3 × 3 convolution kernel on GPU, which makes the reasoning efficiency of 3 × 3 convolution kernel significantly higher than other convolution kernels under the same hardware structure. Secondly, our algorithm structure has more branches than yolov7-tiny, which not only increases the cost of memory access, but also affects the reasoning speed to a certain extent.

5.4.2. Qualitative experiment

In addition, we use several algorithms to visually analyze the detection results on our dataset, as shown in Fig. 20 and Fig. 21. We have selected two images as examples, including two inspection perspectives: user repair perspective (Fig. 20) and robot perspective (Fig. 21). The difference lies in that the perspective of user is higher, so the image is usually from an upright perspective. However, drones are limited by their height, and their viewing angle is often an upward view, which may cause the detected objects to be denser or even overlap visually. At the same time, three classic electricity meter boxes are displayed, including multiple meter boxes and single meter boxes. In order to facilitate the display of labels, we use the form of initials for the labels during the detection process, and the confidence level is expressed after the label name.

In the meter box shown in Fig. 20, the correct detection results should include 1 Multi box_a Normal (bn2), 3 Multi box_p Damage (pd2), and 1 Multi box_p Normal (pn2). The upper right corner viewing window is generally intact, but there are minor cracks that should be classified as damaged, which is the difficulty of this inspection. From Fig. 20, it can be seen that (b) fail to detect damage to the upper right corner viewing window, while (c) miss detection. Figures (a), (d), and (e) all detect correct results, but (e) has a relatively higher confidence level.

In the meter box shown in Fig. 21, the correct detection results should include 1 Multi box_a Damage (bd2), 7 Multi box_p Normal 2 (pn2), 2 Single box_p Normal (pn), and 1 Single box_a Normal (bn). The cover plate in the lower right corner of the meter box has become loose and opened, which may cause water ingress or exposed wires inside the meter box. Therefore, according to the labeling standards, it should be classified as a damaged box; In addition, although the two peep windows in the upper left corner are very close, two peep windows should be detected. The above two points are the difficulties of this detection. From Fig. 21, it can be seen that (a) misdetection occur, (c) two closely spaced peep windows are detected as one, (d) miss detection and the damage to the box is not identified, and (b) and (e) both are correctly detected.

5.5. Generalization verification on public dataset

To verify the universality of our proposed algorithm, we conducted experiments using the publicly available insulator dataset "RSIn-Dataset"(Shuang et al., 2023), which is also used in the power inspection. RSIn-Dataset contains 1887 images and 3286 insulator targets, with image resolution ranging from 1152 × 864 to 7360 × 4912. The experimental results are shown in Table 5. From the TABLE V, it can be seen that our proposed algorithm is still in a leading position. Our algorithm achieved 93.5 % and 55.2 % for mAP_{50} and $mAP_{50:95}$, respectively. They are 0.4 % and 0.3 % higher than the highest values of other algorithms, respectively. This proves that our algorithm can be also suitable to other object detection for power inspection, which has good generalization ability.

6. Conclusion

In this paper, we creatively introduce image recognition technology into the inspection of electric meter box. We collected the image data of the meter box and built a dataset. Subsequently, we made improvements based on YOLOv7-tiny. Finally, we proposed a deep learning structure designed for meter box positioning and defect detection. The proposed EMB-YOLO model focused on the further lightweight of YOLOv7 tiny, which has reduced the model parameters by 20.3 % and improved the FPS by 14.3 % on NVIDIA Titan V. In addition, through the rational use of big kernel convolution, attention and structure reparameterization, the ability of feature extraction and object location of the model has enhanced. The mAP_{50} is increased by 3.5 %, while the $mAP_{50:95}$ is increased by 5.5 %. The experimental results show that the proposed structure can meet the requirements of robustness, accuracy and speed of the meter box defect detection.

The limitations of the proposed algorithm are as follows. Firstly, the large convolutional kernels used in the algorithm have not yet been optimized at the hardware level, which limits its performance when the algorithm deployed on hardware devices, and the detection speed has not been significantly improved. Secondly, our model is relatively lightweight, which limits its ability to extract detailed information from images, resulting in suboptimal performance in small object detection. In future, we will consider how to optimize the large convolutional kernels at the hardware level and how to reduce the number of branches in the network to decrease hardware access frequency. Besides, how to balance the small object detection capability and model efficiency for lightweight models is also an important future work.

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.

Acknowledgements

This work was supported by the Guangxi Science and Technology base and Talent Project (Grant No. Guike AD22080043), the Natural Science Foundation of Guangxi (Grant 2022GXNSFBA035661), Guangxi Science and Technology Program: Guangxi key research and development program (Grant No. Guike AB21220039) and Bagui Scholars Project (Feng Shuang).

References

Aguilar, E., Bolanos, M., Radeva, P., 2019. Regularized uncertainty-based multi-task learning model for food analysis. J. Vis. Commun. Image Represent. 60, 360–370.
Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
Dai, J., Li, Y., He, K., et al., 2016. R-fcn: Object detection via region-based fully convolutional networks. Adv. Neural Inf. Proces. Syst. 29.
Ding X, Zhang X, Han J, et al. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11963-11975.
Duan, N., Huang, C., Sun, C.-C., Min, L., 2022. Smart meters enabling voltage monitoring and control: the last-mile voltage stability issue. IEEE Trans. Ind. Inf. 18 (1), 677–687. https://doi.org/10.1109/TII.2021.3062628.
Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. 88 (2), 303–338.
Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440–1448.
Gao, Z., Yang, G., Li, E.n., Liang, Z., Guo, R., 2021. Efficient parallel branch network with multi-scale feature fusion for real-time overhead power line segmentation. IEEE Sens. J. 21 (10), 12220–12227. https://doi.org/10.1109/JSEN.2021.3062660.

Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580–1589.

He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770–778.

Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314–1324.

Hu J, Shen L, Sun G. Squeeze-and-excitation models[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132–7141.

Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional models[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700–4708.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural models. Commun. ACM 60 (6), 84–90.

Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976, 2022.

Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21–37.

Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116–131.

McLaughlin, S., Holbert, B., Fawaz, A., Berthier, R., Zonouz, S., 2013. A multi-sensor energy theft detection framework for advanced metering infrastructures. IEEE J. Sel. Areas Commun. 31 (7), 1319–1330. https://doi.org/10.1109/JSAC.2013.130714.

Qi, R., Zheng, J., Luo, Z., Li, Q., 2022. A novel unsupervised data-driven method for electricity theft detection in AMI using observer meters. IEEE Trans. Instrum. Meas. 71, 1–10. https://doi.org/10.1109/TIM.2022.3189748.

HAO Shuai, YANG Lei, MA Xu, et al. YOLOv5 transmission line fault detection based on attention mechanism and crossscale feature fusion[J/OL] . Proceedings of the CSEE, 2022 1-12[2022-05-01] . http://kns.cnki.net/kcms/detail/11.2107.tm.20220126.1718.008.html (in Chinese) .

Shuai, H.A.O., Ruize, M.A., Xinsheng, Z.H.A.O., et al., 2021. Fault detection of YOLOv3 transmission line based on convolutional block attention model. Power System Technol. 45 (8), 2979–2987 in Chinese.

Shuang, F., Chen, X., Li, Y., Wang, Y., Miao, N., Zhou, Z., 2022. PLE: Power Line Extraction Algorithm for UAV-Based Power Inspection. IEEE Sensors J. 22 (20), 19941–19952. https://doi.org/10.1109/JSEN.2022.3202033.

Shuang, F., Han, S., Li, Y., Lu, T., 2023. RSIn-dataset: an UAV-Based Insulator Detection Aerial Images Dataset And Benchmark. Drones 7 (2), 125.

Simonyan K, Zisserman A. Very deep convolutional models for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1–9.

Takiddin, A., Ismail, M., Zafar, U., Serpedin, E., 2021. Robust electricity theft detection against data poisoning attacks in smart grids. IEEE Trans. Smart Grid 12 (3), 2675–2684. https://doi.org/10.1109/TSG.2020.3047864.

Takiddin, A., Ismail, M., Serpedin, E., 2023. Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids. IEEE Trans. Smart Grid 14 (1), 663–676. https://doi.org/10.1109/TSG.2022.3193989.

Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural models [C]//International conference on machine learning. PMLR, 2019: 6105-6114.

Tao, X., Zhang, D., Wang, Z., Liu, X., Zhang, H., Xu, D., 2020. Detection of power line insulator defects using aerial images analyzed with convolutional neural models. IEEE Trans. Syst. Man Cybernet. Syst. 50 (4), 1486–1498. https://doi.org/10.1109/TSMC.2018.2871750.

Wandt B, Rosenhahn B. Repnet: Weakly supervised training of an adversarial reprojection model for 3d human pose estimation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 7782–7791.

Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534–11542.

Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 7464–7475.

Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]// Proceedings of the European conference on computer vision (ECCV). 2018: 3–19.

Xia, X., Xiao, Y., Liang, W., Cui, J., 2022. Detection methods in smart meters for electricity thefts: a survey. Proc. IEEE 110 (2), 273–319. https://doi.org/10.1109/JPROC.2021.3139754.

Xu, C., Li, Q., Zhou, Q., Zhang, S., Yu, D., Ma, Y., 2022. power line-guided automatic electric transmission line inspection system. IEEE Trans. Instrum. Meas. 71, 1–18. https://doi.org/10.1109/TIM.2022.3169555.

Yang, L., Fan, J., Xu, S., Li, E.n., Liu, Y., 2022. Vision-based power line segmentation with an attention fusion network. IEEE Sensors J. 22 (8), 8196–8205. https://doi.org/10.1109/JSEN.2022.3157336.

Zidi, S., Mihoub, A., Mian Qaisar, S., Krichen, M., Abu Al-Haija, Q., 2023. Saeed Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. J. King Saud Univers. Comput. Inform. Sci. 35 (1), 13–25.