Application of GCB-Net based on Defect Detection Algorithm for Steel Plates

BeiBei Fan (fanbeibei@shu.edu.cn)
Shanghai University
Weixin Li
Shanghai University

Research Article

Keywords: Surface Inspection, Defect Recognition, Lightweight network, Attentional Mechanism, YOLOV4

Posted Date: April 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1550068/v1

Application of GCB-Net based on Defect Detection Algorithm for Steel Plates

*BeiBei Fan, Weixin Li

Abstract

In recent years, deep learning convolutional networks have been gradually applied to metal surface inspection. Steel plate defect detection helps improve the efficiency of steel plate production by locating and classifying surface defects on steel plates. The related methods are challenging to balance the accuracy of localization and classification. There are scenarios in the industrial inspection field where the data set is relatively small, and the distribution of defects is significantly irregular; this paper extends the traditional image detection network by introducing an attention mechanism. At the same time, to meet the requirement of higher real-time performance in industrial inspection, the lightweight Ghost network is introduced to replace the original backbone network CSPDarknet53 in YOLOV4 to improve the detection speed of the network. This paper constructs the high accuracy and low latency defect detection algorithm Ghost-CBAM-YOLOV4 Network (GCB-Net). The GCB-Net model is based on the improved YOLOV4 target detection network. The CBAM module and Ghost lightweight network are introduced to complete the prediction of defect categories without reducing the classification and detection accuracy. The desired experimental results are achieved, which is essential for detecting defects in steel plates and improving product quality.

Keywords Surface Inspection; Defect Recognition; Lightweight network; Attentional Mechanism; YOLOV4

1 Introduction

For industrial products, the quality of the product surface will directly impact the product's performance indicators and final value, and the surface quality standards are rigorous. According to statistics, surface defects account for 95% of certain process products. High precision and efficient surface inspection means can provide reliable feedback for closed-loop control of shop floor production, so it is essential to study advanced methods for detecting product surface defects.

BeiBei Fan

fanbeibei@shu.edu.cn

Weixin Li

li_weixin@shu.edu.cn

School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China

The steel industry is currently in the overall overcapacity and insufficient capacity for high-end products. Among them, the qualification rate of high value-added steel is only 80%, in which performance defects are common product defects. Traditional feature extraction based on manual experience, using quality detection methods based on image processing, is "feature extraction and classification." For example, algorithmic classifiers: Support Vector Machine (SVM) [1] Scale Invariant Feature Transform (SIFT) [2], these results rely on manual a priori knowledge and lack generalizability. These methods are affected by the environment, lighting, and accuracy, can only be used in specific scenarios, and are not suitable for general detection environments [3], nor are they capable of complex multi-target detection [4-7]. Whereas deep learning methods can extract features from complex environments for multiple targets to cope with the small dataset of hot rolled steel surface defects, some papers have used semi-supervised learning to expand [8, 9]. Some use GAN networks to generate pseudo-labels [10], and these methods achieve supervised learning by using a large amount of unlabelled data. Some scholars proposed end-to-end detection by feature fusion [11], and most of these steel plate defect detection studies relied on the NEU-DET dataset from Northeastern University [12].

This paper aims to investigate a more pre-industrial mode, generalized real-time defect detection method for steel plates. The NEU-DET dataset from Northeastern University collects six types of defects, Inclusions, Scratches, Rolled-in_scale, Crazing, Pitted Surface, and Patches, with 300 images of each defect and 200×200 image size. Ensuring the accuracy of detection and real-time rate when the data samples are not large will increase the difficulty and challenge of localization and classification.

The main contributions of this paper to this work are threefold:

1) In this paper proposes a more straightforward end-to-end steel defect detection method, GCB-Net, based on the existing theoretical basis and the actual production environment. By choosing YOLOV4 [13], which has higher accuracy and speed, as the base network, and introducing Ghost [14], a lightweight network, this more lightweight network structure will be more suitable for the online identification task of steel plate defect detection.

2) The attention module Convolutional Block Attention Module (CBAM) [15] is introduced on top of the three adequate feature layers extracted from the original YOLOV4 backbone network. CBAM is an attention mechanism module that combines spatial and channel to enhance the fusion of features and improve the discriminative power of the network to improve the classifier performance and improve the accuracy and recognition of steel surface defects.

3) The improved network GCB-Net is compared with traditional networks such as YOLOV4, Faster-RCNN [16], MobileNet [17, 18], and the same NEU-DET dataset from Northeastern University is used for validation. The GCB-Net proposed in this paper achieves a detection speed of 66.14 fps and detection accuracy of 82.78% mAP, which exceeds the general detection network and obtains more desirable results.

2 Related Work

In the past two years, many scholars have done investigations into steel plate defect detection. As it is tough to collect many balanced samples of various surface defects, scholars mainly focus on improving the accuracy and rate of industrial defect detection, sample imbalance, and other issues related to several aspects of research work.

2.1 Deep Learning for Steel Surface Defect Detection

Research on Sample Imbalance in Industrial Datasets

Industrial defect datasets often suffer from the problem of insufficient samples, and scholars have tried to make full use of a large number of images with unlabeled defect types to mine the data features in unlabeled data and thus improve the classification effect. Yu He first proposed a method based on multi-training and generating adversarial networks, using GAN to generate adversarial networks and residual networks by generating unlabeled samples, and then proposed a multi-training algorithm based on two learners with different learners, obtaining 99.5% classification accuracy [10], Recently, they proposed a semi-supervised learning method to improve CNNs by pseudo-labeling, achieving an accuracy of 90.7% with limited labeled data [8]. A CAE-SGAN approach has also been proposed to solve the problem of insufficient samples by training convolutional autoencoders (CAE), using the encoder network of CAE as a feature extractor and Semi-Supervised Learning with Generative Adversarial Networks (SGAN) to introduce semi-supervised learning and create pseudo-labels to further improve the generalization ability and the classification accuracy of hot-rolled steel by 16% [5]。

Research on Integrated Deep Convolutional Neural Networks

Many scholars build deeper networks with better feature extraction modules to better accomplish the work of defect detection. Integrated learning increases the stability of classification and overcomes the overfitting problem caused by small samples through the integration of multiple learners, thus improving classification accuracy. Hongwen Dong extracts multi-scale features from the backbone network by pyramid feature fusion and global contextual attention network (PGA-NET), fuses the features into five resolutions using the pyramid feature fusion module, and applies the fused feature map based on the contextual attention module to adjacent resolutions, obtaining 82.15% accuracy on NEU-SEG [19]. Hao, Rui yang proposed a feature fusion network based on a balanced feature pyramid, in which deformable convolution allows the convolution to fit the defect shape better to generate high-quality, multi-resolution feature maps to help detect defects of multiple sizes [20]. Some scholars have proposed a fused multilayer-style end-to-end detection method by generating feature maps at each stage through CNN, using a multi-level feature fusion network (MFN) to fuse multiple layered features into one. A region proposal network (RPN) to generate regions of interest (ROI), using only 50 proposals on NEU-DET, 20ft/s detection speed on a single GPU [11]. To extract richer multi-level defect features, a coded residual network for significant target detection (EDR-NET) has been proposed to accelerate the

convergence of the model by fusing attention mechanisms [21]. Some scholars have used a spectral clustering super-pixel segmentation algorithm to obtain better accuracy based on significant texture defect features by using multiple constraints and exploiting features [22]. However, these integrated learners reduce the detection rate while increasing the accuracy due to the complexity of the network model.

A visual inspection system has also been proposed for defect detection, consisting of two main components. The first is an optical system design technique that produces images that easily distinguish between defective and non-defective areas. The second is an automatic detection algorithm technology that can accurately detect the location and shape of defects in the images generated from the optical system [23].

2.2 Research on Attentional Mechanisms

The visual attention mechanism is a brain signal processing mechanism unique to human vision. By quickly scanning the global image, human vision automatically acquires the target area to be focused on, which is generally referred to as the focus of attention, and then devotes more attentional resources to this area to obtain more detailed information about the target to be focused on and to suppress other useless information. Attention mechanisms have been used in various fields, such as computer vision and natural language processing [24, 25]. They have been widely used in sequential models, using recurrent neural networks and long-term short-term memory (LSTM). Various forms of models of attention mechanisms have been continuously proposed, for example, Channel attention mechanism (Channel attention) [26], Spatial Attention (Spatial attention) [27], Semantic Attention (Semantic attention) [28], Multi-layer Attention) [15, 29]. The more classical and commonly used ones are the channel and spatial attention mechanisms.

3 Proposed Method

To accomplish the task of industrial steel defect detection more accurately and in real-time, this paper proposes a network architecture named Ghost-CBAM-YOLOV4 Network (GCB-Net). As shown in Figure 1 below, YOLOV4 is chosen as the base network. By modifying Backbone, a lightweight network, Ghostnet is introduced, which will reduce the overall number of parameters of the model and thus improve the overall speed of the model. Meanwhile, after the original YOLOV4 backbone network extracts three adequate feature layers through the CBAM module, it will first pass through a channel attention module. After getting the weighted results, it will then pass through a spatial attention module. The final weighting is performed to get the results to model the correlation between feature channels, and the essential features are reinforced to improve the accuracy. The experiments show that we can obtain more satisfactory enhanced feature extraction results.

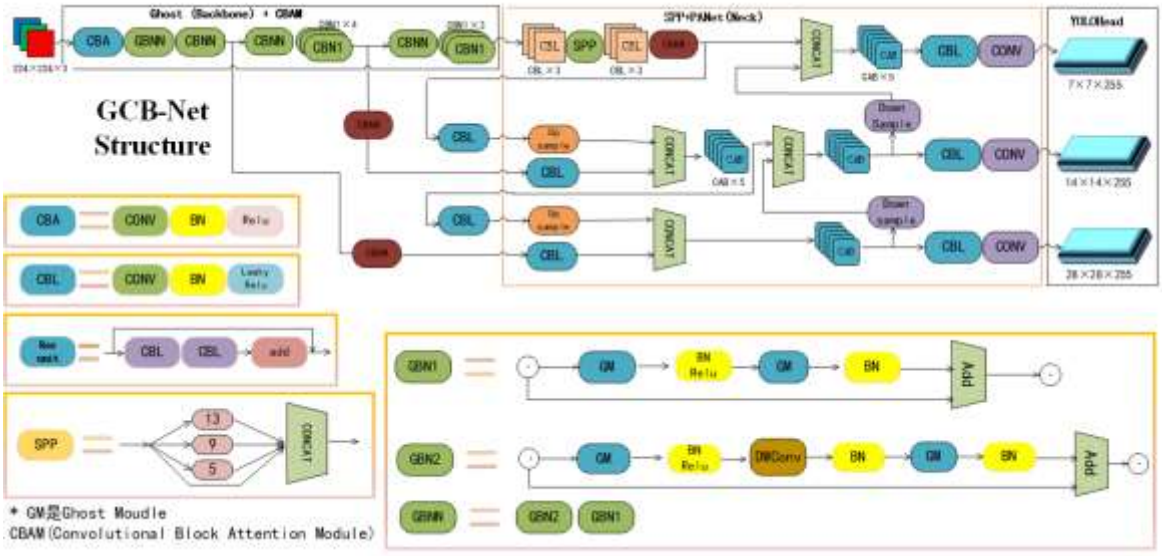
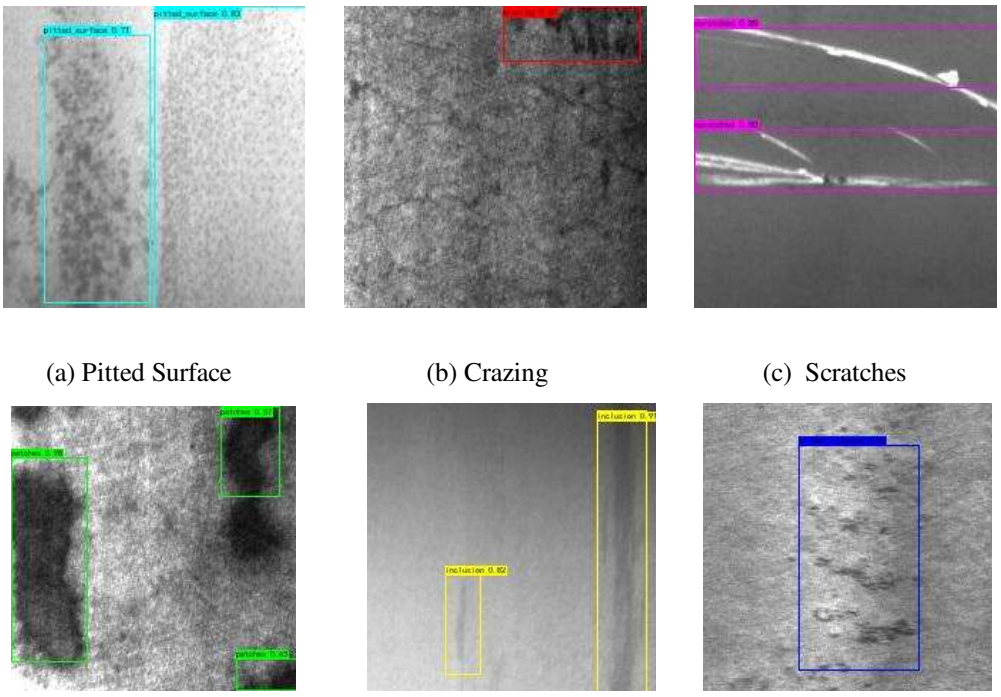


Fig.1 GCB-Net network structure diagram. GCB-Net is divided into three parts, Ghost network as Backbone, SPP, and PANet as Neck part, and the output is three YOLO Head.

3.1 Motivation

This paper relies on the actual production of industrial scenarios. It collects six common types of steel surface defects, which are Pitted Surface, Crazing, Scratches, Patches, Inclusion, and Rolled-in_scale, as shown in Figure 2 below:



(a) Pitted Surface (b) Crazing (c) Scratches

(d) Patches (e) Inclusion (f) Rolled-in_scale

Fig. 2 The examples of test images in NEU-DET. The dataset contains a total of six defects: (a) Pitted Surface, (b) Crazing, (c) Scratches, (d) Patches, (e) Inclusion, (f) Rolled-in_scale.

This dataset contains 300 of each defect. The image size is 200×200. Due to the small size of the image, the accuracy is low, which increases the difficulty of detection. The original image dataset is small. Making the model fit better will become the key to improving the accuracy. From Fig (a), (c), (d), and (e), it can be seen that a picture may contain many different defects at different locations, and an image may also have different kinds of defects. In this paper, we will use GCB-Net to complete the classification and localization tasks of this steel defect detection dataset.

3.2 Model

Yolov4 Algorithm Analysis

In this paper, the Yolov4 algorithm is chosen as the baseline network, which is based on the C language, and CUDA's Darknet framework has good compatibility portability and is much faster than the deep neural network algorithm based on candidate regions so that it can meet the demand of real-time industrial defect detection. From 2016, when YOLOV1 was first proposed, to 2020, YOLO4 was proposed, spanning four generations. Compared with the previous version, the YOLOV4 algorithm improves the detection accuracy while maintaining a higher detection speed and is more than 10% ahead of YOLOV3 [30]in terms of both accuracy and speed. YOLOV4 has the basis of single-core GPU operation in terms of hardware configuration and has made innovative contributions to target detection in terms of hardware cost.

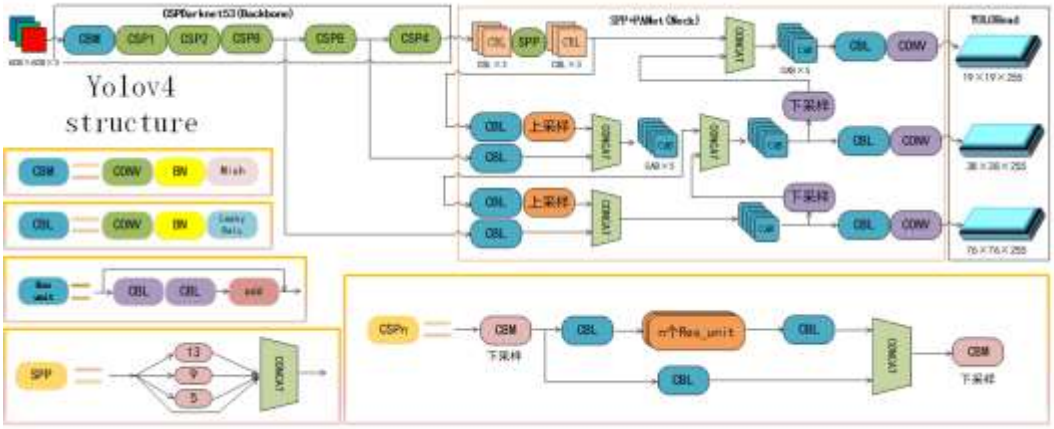


Fig. 3 YOLOV4 network structure diagram. YOLOV4 is divided into three parts, CSPDarknet network as Backbone, SPP, and PANet as Neck part, and the output is three YOLO Head.

Figure 3 shows the network structure of YOLOV4. It can be seen in the figure that the input is computationally processed in the backbone layer to obtain an accurate feature map. The obtained multiple feature maps are further processed in the Neck layer. The nonlinear feature maps processed by the Neck layer are finally sent to the Head layer for processing, where the final detection results of the target can be obtained.

The framework of YOLOV4 is relatively simple, and its hierarchy is very visual. It represents the three parts that make up a "human," Backbone, Neck, and Head, as shown in Figure 4 below:

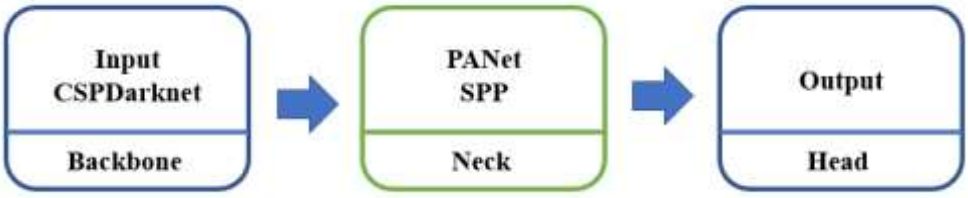


Fig. 4 Sketch of YOLOV4 framework. The structure diagram is mainly divided into Backbone, Neck, and Head.

The functions of each section are as follows:

1) Backbone: Various convolutional networks, the purpose of which is to do preliminary feature extraction of the original image, to solve the problem of excessive computational effort that may occur in the forward inference process due to duplication of information. The main effect of Yolov4 using CSPNet is to reduce the computational effort and have good expressiveness for rich gradient combinations.

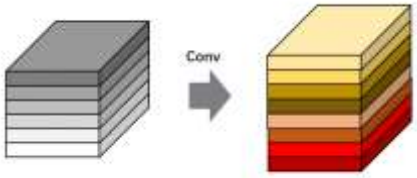
2) Neck: The purpose of various structures is to do "feature fusion" on the structure, mainly to solve the problem of small target detection, overlapping target detection, etc. Different from Yolov3, Yolov4's Neck layer is also a significant feature, and the process of pooling uses Spatial Pyramid Pooling (SPP). SPP pooling is mainly for the feature values in multiple dimensions to obtain various pooled feature values. Finally, these different feature values are stitched together.

3) Head: The Head layer of Yolov4 mainly incorporates three parts: output loss calculation and prediction result parsing. The output primarily uses two layers of convolution operations to output images of the required size for the experiment, which is also the actual output of the network. The loss calculation includes calculation of classification, confidence interval calculation, rectangular box position and loss calculation of width and height, and output after weighted summation. The prediction process mainly uses the corresponding interface of the test branch. It uses the Sigmoid operation to transform the position x and y prediction between 0 and 1 by traversing all the pictures output in the calculation results. At the same time, the predicted confidence interval and category score are obtained, the part of the confidence interval with a certain threshold is reserved for the remaining feature maps, and the final prediction result is output.

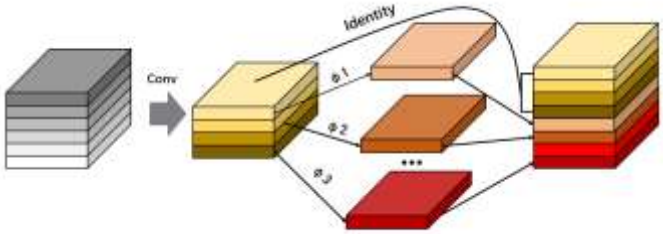
Ghost Module

In 2020, Huawei proposed a new lightweight network and named it GhostNet. To solve the problem of redundancy in feature maps, it is proposed to consider similar feature maps as ghosts of each other and use some operations with lower computational complexity (Cheap Operations) to generate these redundant feature maps. These operations can reduce the number of parameters of the model and improve the execution speed of the model under the condition of ensuring a good detection effect. A module named Ghost Module is designed in GhostNet. Its function is to replace ordinary

convolution. The difference between standard convolution and Ghost Module is shown in Figure 5 below:



(a) Ordinary convolution



(b) Ghost Module

Fig. 5 The difference between ordinary convolution and Ghost Module. (a) is Ordinary convolution, (b) is Ghost Module.

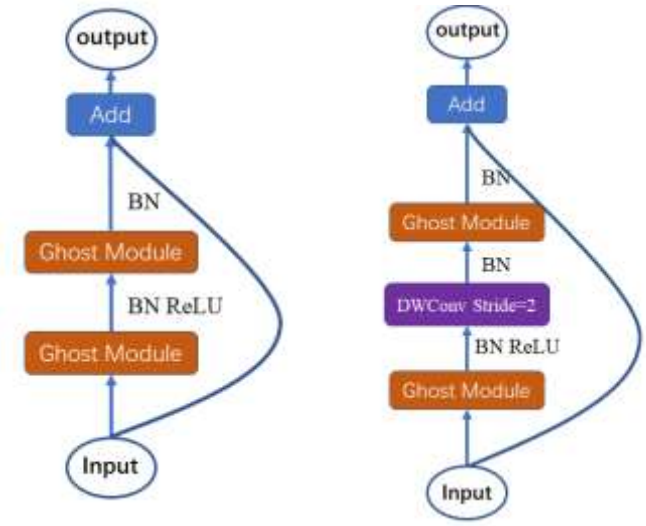
The Ghost Module divides the standard convolution into two parts and first performs an ordinary 1x1 convolution, which is a small amount of convolution. For example, a 32-channel convolution is typically used, and a 16-channel convolution is used here. The function of this 1x1 convolution is similar to feature integration, generating feature concentration of the input feature layer and then performing depthwise separable convolution. This depthwise separable convolution is a layer-by-layer convolution that generates Ghost feature maps by utilizing the feature enrichment obtained in the previous step.

Ghost Bottlenecks is a bottleneck structure composed of Ghost Modules, as shown in Figure 6 below, which essentially uses Ghost Modules to replace ordinary convolutions in the bottleneck structure. Ghost Bottlenecks can be divided into the central part and the residual edge part, including the Ghost Module, which we call the central part.

There are two types of Ghost Bottlenecks, as shown in Figure 6 below, figure 6(a) stride=1. The trunk channel is composed of two Ghost Modules (GM) in series, where the first GM expands the number of channels, and the second GM expands the channel number. The number is reduced to match the number of input channels. The skip connection path is used similarly to ResNet [31]. In this way, the input and output dimensions of Ghost-BottleNeck are also consistent, which can be easily embedded into other CNN networks.

The difference in Figure 6(b) is that a Deepwise convolution with stride=2 is added between the two GMs of the main channel, this structure can reduce the size of the

feature map to 1/2 of the input, and the same downsampling is also required for the skip connection path to ensure that the Add operation can be aligned. This module can replace downsampling layers (1/2) in other CNNs.



(a) Stride=1 bottleneck (b) Stride=2 bottleneck

Fig. 6 Bottleneck with different lengths. (a) is Stride=1 bottleneck, (b) is Stride=2 bottleneck.

Convolutional Block Attention Module

Convolutional Block Attention Module (CBAM) sequentially applies channel and spatial attention modules to learn what and where to pay attention in the channel and spatial dimensions. The structure of CBAM is shown in Figure 7 below:

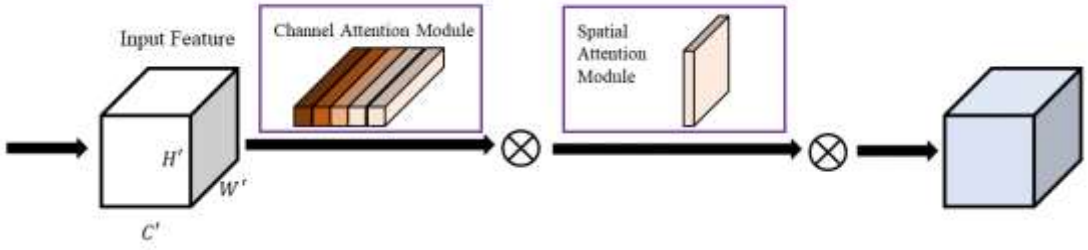


Fig. 7 Convolutional Block Attention Module. Input Feature first passes Channel Attention Module and then through Spatial Attention Module.

As shown in Figure 7, when a feature map is an input, CBAM sequentially derives attention maps along two independent dimensions of channel and space and then multiplies the attention maps into the feature map for adaptive feature refinement. CBAM mainly consists of two parts:

1) Channel Attention Module:

As shown in Figure 8 below, to effectively calculate the channel attention, the spatial dimension of the feature map needs to be compressed, and for the aggregation of spatial information, the commonly used method is average pooling. However, some scholars

believe that max-pooling collects another essential clue about unique object features, which can infer more detailed channel attention, so average pooling and max pooling can be used simultaneously.

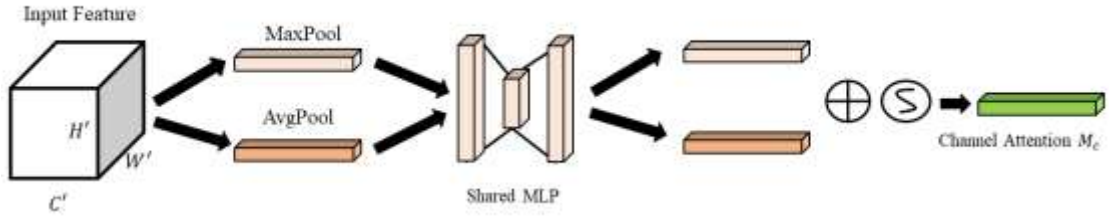


Fig.8 Channel Attention Module for CBAM. The input feature maps are passed through global max pooling and global average pooling, respectively, and then through MLP. Perform an element-wise addition based on the output of the features by the MLP, go through the sigmoid activation operation to generate the final channel attention feature map.

Among them:

$$M_c(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right)$$
$$= \sigma(W_1(W_0\left(F_{avg}^c\right) + \left(W_1\left(W_0\left(F_{max}^c\right)\right)\right) \quad (1)$$

F_{avg}^c and F_{max}^c represent average pooling and max pooling features, respectively. Afterward, the two feature descriptions are forwarded to a shared network for generating channel attention maps M_c, The shared network consists of a multilayer perceptron (MLP), It contains a hidden layer, and the size of the hidden layer is set to $R/C = r \times 1 \times 1$, where R is the drop rate. After applying the shared network to the average pooling and max pooling descriptors, the output feature vectors are merged element-wise, σ represents the sigmoid function,$M_c(F)$ multiply with the input feature F to get F'.

2) Spatial Attention Module:

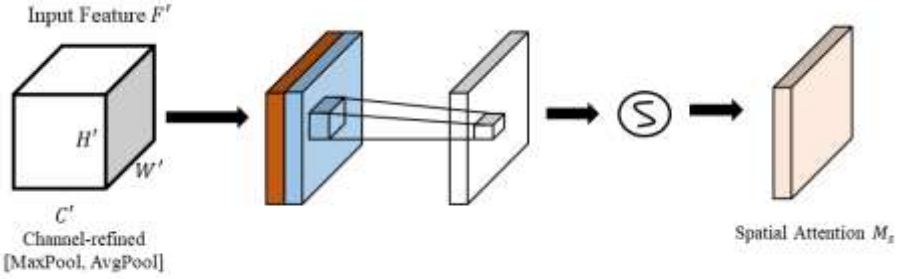


Fig. 9 Spatial Attention Module of CBAM. First, do a channel-based global max pooling and global average pooling, and then do the concat based on the channel. After convolution, the dimension is reduced to one channel. Then generate spatial attention feature through sigmoid. Finally, multiply the feature with the input feature of the module to get the final generated feature.

The spatial attention module focuses on "where" the most informative part, which is complementary to channeling attention. Average pooling and max pooling are applied along the channel axis and then concatenated to generate an efficient feature descriptor to compute spatial attention. A convolutional layer is then applied to generate a spatial attention map $M_s(F)$ of size R×H×W, which encodes the locations that need attention or suppression.

$$M_s(F) = \sigma\left(f^{7\times7}([AvgPool(F); MaxPool(F)])\right)$$

$$= \sigma\left(f^{7\times7}\left([F^s_{avg}; F^s_{max,}]\right)\right) \qquad (2)$$

From Equation 2, it can be seen that the information channel of a feature. Two aggregated by the pooling operation, and two 2D maps are generated, F^s_{avg} size is $1\times$ H $\times$ W, F^s_{max} size is 1×H×W， σ represents the sigmoid function, $f^{7\times7}$ represents a convolution operation with a filter size of 7×7.

One of the significant advantages of CBAM is that it can be easily integrated into existing networks to improve network performance at a low cost and can be trained end-to-end with the basic CNN.

Ghost-CBAM-YOLOV4 Network

In this paper, the lightweight network Ghostnet is innovatively used as the backbone network of Yolov4 for feature extraction, and the CBAM attention mechanism module is added to the three adequate feature layers extracted from the backbone network, which can effectively reduce the amount of parameters and improve the effect of feature extraction. Figure 10 shows a simplified diagram of the GCB-Net network structure:

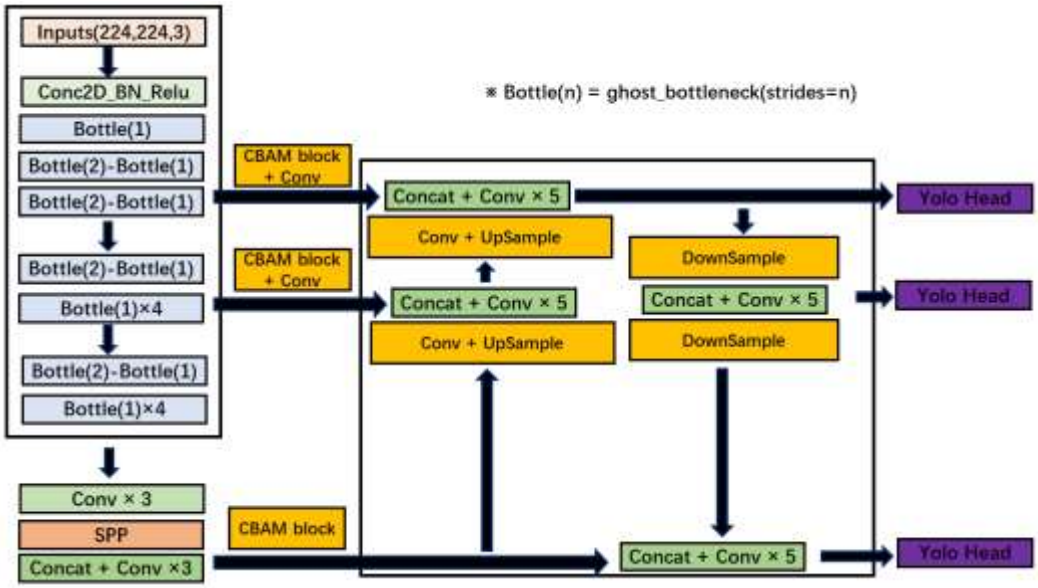


Fig. 10 GCB-Net network structure diagram. The backbone feature network enhanced feature extraction network, and Head are introduced in detail. It can be seen intuitively that the CBAM module is added to the three adequate feature layers extracted.

Since Ghostnet is to be applied to YOLOV4, it can be seen from the figure that when inputting a picture with a size of 224×224×3, first perform convolution, normalization, and Relu activation function, and then send the feature map to ghost_bottleneck, ghost_bottleneck contains two kinds of minister strides, two and one. When strides are two, the height and width will be compressed. For yolov4, we need to use the three effective features obtained by the backbone feature extraction network to strengthen the construction of the feature pyramid. After introducing Ghostnet to complete the feature extraction of the bottleneck structure several times, the feature layer will be taken out three times to construct the (Feature Pyramid Network) FPN network.

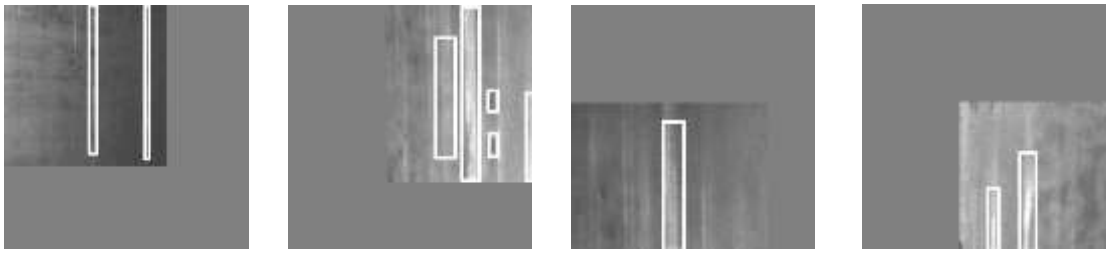
In the backbone feature extraction network composed of Ghost, the CBAM module is added to the three effective feature layers extracted respectively. As shown in Figure 10, it is counted three times in total to better fuse feature information and enhance feature extraction.

3.3 Data Augmentation

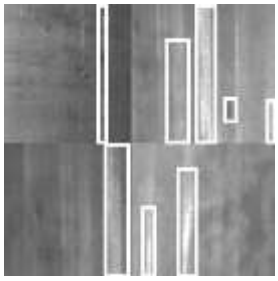
Data augmentation can reduce the imbalance of the data and improve the accuracy of the detection network. This paper adopts four data augmentation methods for Pitted Surface, Crazing, Scratches, Patches, Inclusion, and Rolled-in_scale: 1) Color normalization, take luminance filtering 2) Random crop 3) Random flip, through randomly flipping the original image horizontally or vertically 4) Image is scaled and distorted for length and width.

The methods of data enhancement are divided into online enhancement and offline enhancement. Offline enhancement means that we perform data enhancement on a dataset, save it locally to form a new dataset, and use this new dataset for training. At this point, the locally saved dataset has changed. This paper adopts the method of online generation enhancement during training; that is, data enhancement is performed while training, and the locally saved dataset file will not change. The above data enhancement operation is performed when each image is trained. Since each data enhancement is random, the data set of each training is different from the data set of the previous training, which is equivalent to increasing the amount of data in the dataset.

Yolov4 also includes mosaic data enhancement. As shown in Figure 11 below Mosaic method reads four pictures at a time, flips, zooms, changes the color gamut of the four pictures, scales the input pictures, and arranges them in four directions. After the combination of images and the combination of boxes, it is sent to the network, and the data of the four pictures will be calculated at once when the BN is calculated. This method enriches the background of detected objects and improves the model's fitting ability and generalization ability.



(a) Read four defect pictures at one time and place them in four directions



(b) Divide the pictures and put them together

Fig. 11 Mosaic data augmentation. Combine four pictures into one picture.

4 Experiments

4.1 Experimental Preprocessing

1) The deep metric learning steel plate defect detection algorithm designed in this paper is based on the Pytorch deep learning open-source framework, which provides a complete toolkit for training, testing, fine-tuning, and developing models. The operating environment of the computer used in this study adopts the Linux operating system, and the computer software and hardware environment table are shown in the following table 1:

Table 1 Computer configuration parameters. Table contains software configuration and hardware configuration.

Configuration Information	Parameter
CPU	i7-8700k
GPU	NVIDIA GTX1080Ti
Deeping Learning Framework	Pytorch
RAM	32G
Number of Virtual Threads	12

2) The training is divided into two phases, the freezing phase and the unfreezing phase. In the freezing stage, the Backbone of the model is frozen, the feature extraction network does not change, the memory occupied is small, only the network is fine-tuned, the training is set to 50 epochs, and the freezing learning rate is 1e-3; In the training phase of the thawing stage, the Backbone of the model is not frozen, the feature extraction network will change, the memory occupied is large. All the parameters of the network will be changed. The thawing learning rate is 1e-4, and the training is 50 epochs. The total is 100 epochs, and the weight is reserved for every other epoch.

When the training is completed, 100 weight files will be obtained, and the weight file with the smallest val_loss will be selected and imported for prediction.

3) The segmentation of the Northeastern University dataset NEU-DET is as follows:

Table 2 NEU-DET dataset partition. The table contains categories, train, val, and test numbers.

	Categories	Train	Val	Test
NEU-DET	6	1440	180	180

As mentioned above, this paper adopts the online enhancement method, and the online enhancement data method includes four enhancement transformations and Mosaic enhancement. A total of 100 epochs, of which 50 are frozen epochs, each epoch training data set is 91, 50 thawed epochs, each epoch training data set is 183, the size of the input image is resized to 224 × 224 × 3.

4.2 Evaluation Metrics

This paper will use Recall, Precision, F1-Score, Average Precision (AP), Mean Average Precision (mAP), and Frames Per Second (fps) to evaluate the metrics to evaluate the proposed method. Before understanding precision and recall, we introduce the concept of the Confusion Matrix.

1) Confusion matrix is generally used for classification, and its matrix table is shown in Table 3 below:

Table 3 Confusion matrix. In the form of a matrix, the records in the data set are summarized according to the two criteria of the real category and the category predicted by the classification model.

True label / Pred label	Positive	Negative
Positive	TP	FN
Negative	FP	TN

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{F1} - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

It can be seen from Table 3 that TP, FN, FP, and TN represent true positive, false negative, false positive, and true negative, respectively. Obtaining these four indicators through calculation, Precision, Recall, and F1-Score can be obtained through formulas (3), (4), (5). Use F1-Score to comprehensively evaluate the performance of Recall and Precision.

2) Intersection Over Union (IOU) It can be seen from formula 6 that the calculation of IOU is obtained by comparing the intersection of the real frame and the predicted frame with its union. IOU is only used to measure monitoring accuracy and is limited to a specific data set, so IOU can measure any prediction range obtained in the output. Its mathematical formula 6 is as follows：

$$IoU_{AB} = \frac{\text{A} \cap \text{B}}{\text{A} + \text{B} - \text{A} \cap \text{B}} \tag{6}$$

Among them: A -- Selection range of the real box

B -- Selection range of the prediction box

3) Average Precision (AP), when measuring the model's performance, simply using Precision and Recall cannot fully reflect the accuracy of its performance. By using the area under the Precision curve as a measure, we have the concept of AP value. The calculation method of AP is 11-point interpolation average precision, which calculates the average value of the maximum precision of 11 recall levels. These 11 points are sorted by confidence, from 0.0 to 1.0. The mathematical expression is shown in the following formula 7:

$$AP = \frac{1}{11} \sum_{r \in \{0.0,\ldots,1.0\}} AP_r \qquad (7)$$

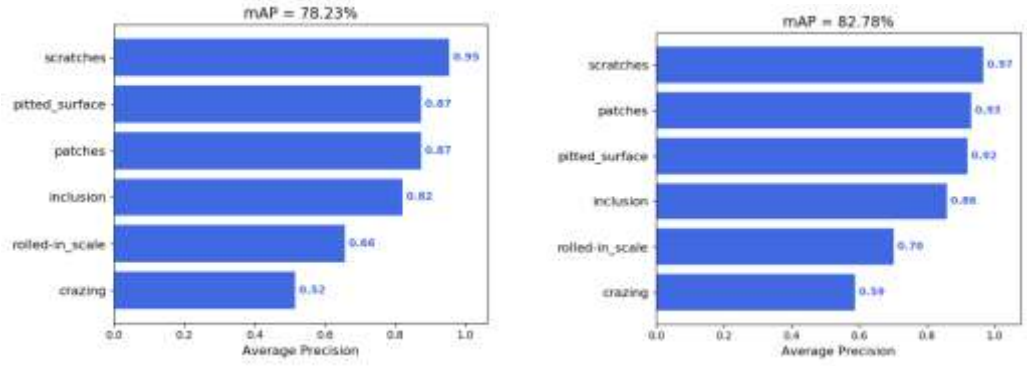
4) Mean Average Precision (mAP), mAP is the average of AP. The calculation of mAP is calculated on a certain data set. If the AP measures the pros and cons of the training model in each category, then the mAP measures the pros and cons of the training model in all categories.

5) Frames Per Second (FPS), The frame rate is defined as the refresh rate of the continuously broadcast image data on display. The frame rate can also be called the frame frequency, and the unit is Hz. It shows the speed with which the whole training process is completed and the smoothness of the image playback.

4.3 Experimental Results

Ablation experiments are performed by combining GCB-Net and traditional YOLOV4, Faster-RCNN, a lightweight network Mobilenet with a depthwise separable convolution module. The effects of introducing Ghost Module and CBAM on the final detection performance are analyzed. The experimental results are divided into ABCDE and shown as follows:

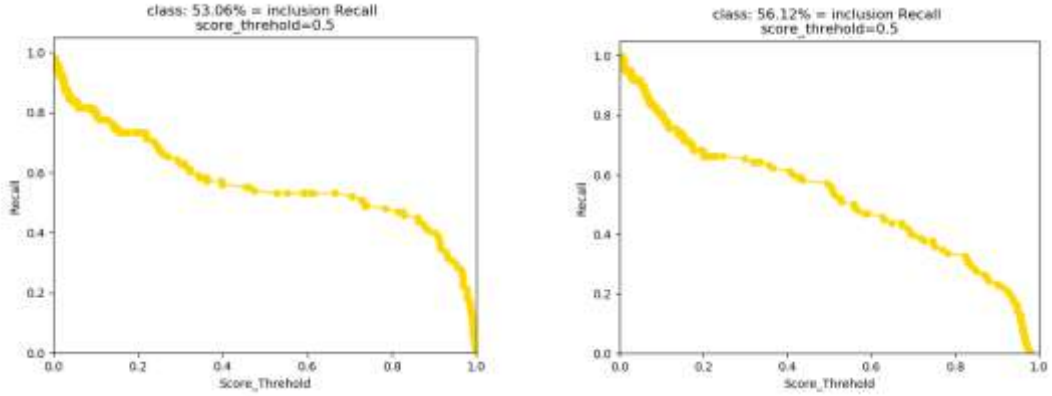
A) After training, by comparing our method (GCB-Net) to YOLOV4, using AP to test each category and mAP to evaluate the dataset, As shown in Figure 12 below:



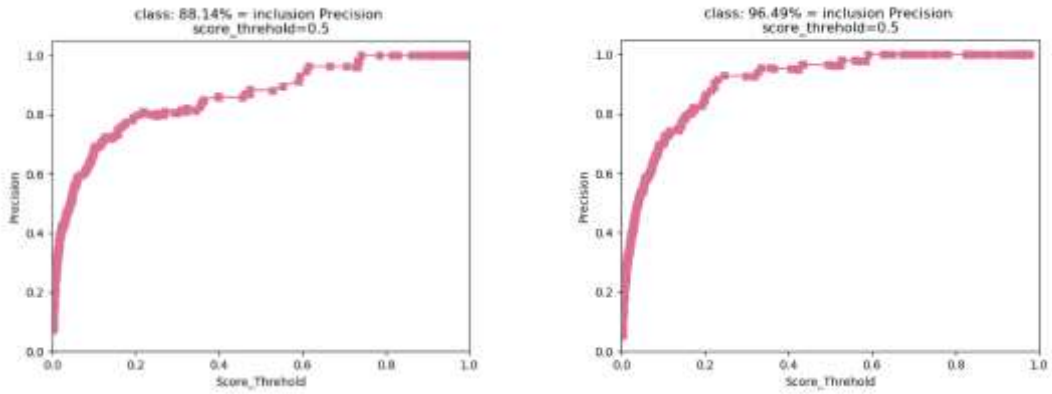
(a) YOLOV4 test results (b) GCB-NET test results

Fig. 12 mAP comparison between YOLOV4 and GCB-Net. Fig contains six categories of AP and mAP for the total dataset.

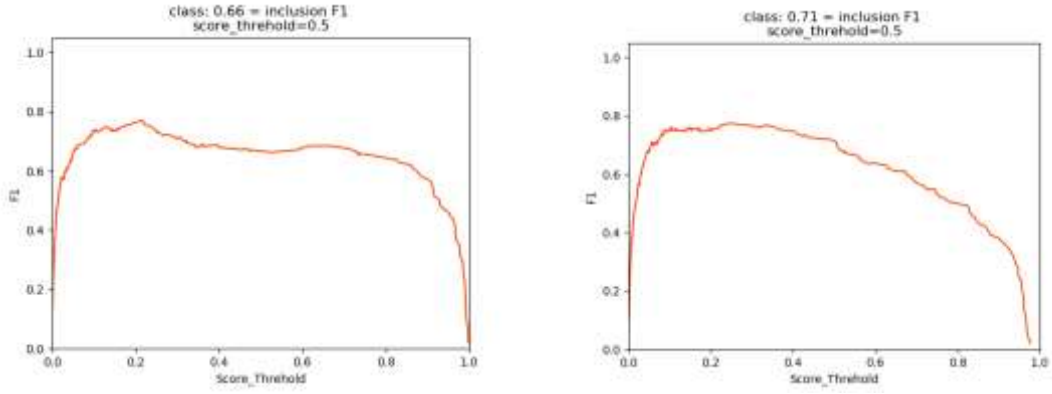
As shown from the above figure 12, the mAP value of GCB-Net is 82.78, which is 4.55 percentage points higher than that of YOLOv4. It can be seen from the AP values of various types that Inclusion is one of the categories with a relatively large increase. The following will select the Inclusion category to evaluate it for Precision, Recall, and F1-score, as shown in Figure 13 below:



(a) Recall comparison chart of YOLOV4 and GCB-Net



(b) Precision comparison chart of YOLOV4 and GCB-Net



(c) Comparison of F1-score of YOLOV4 and GCB-Net

Fig. 13 Comparison of Precision, Recall, F1-score between YOLOV4 and GCB-Net.

The score_threhold is an adjustable hyperparameter, and the same score_threhold is used for this experiment. Through the comparison of Precision, Recall, and F1-score, it

can be seen that GCB-Net has achieved better results on the defect of Inclusion, which shows that the model proposed in this paper has better detection performance.

B) Before and after the improvement of this experiment, the IOU is set to 0.5 to read its performance indicators. The experimental results are shown in Table 4. The unit of mAP is a percentage, and the AP of various categories directly displays the counting ratio with one as the total score:

Table 4 Detection Results mAP and AP on NEU-DET.

Method	mAP	crazing	inclusion	patches	Pitted surface	Rolled-in scale	scratches
YOLOV4	78.23	0.52	0.82	0.87	0.87	0.66	0.95
Faster-RCNN	41.12	0.16	0.55	0.80	0.12	0.48	0.36
MobileV1-YOLOV4	78.95	0.55	0.81	0.91	0.83	0.66	0.97
MobileV3-YOLOV4	77.88	0.48	0.79	0.88	0.85	0.73	0.95
Ghost-YOLOV4	79.95	0.45	0.83	0.93	0.90	0.72	0.97
GCB-Net (our methods)	82.78	0.59	0.87	0.93	0.92	0.70	0.97

As shown in the table above, the following three conclusions can be drawn:

1) The detection effect of introducing the traditional lightweight MobileNet [30, 31] network is not as good as the detection accuracy of Ghostnet introduced in this paper. The mAP of Ghost-YOLOV4 is 0.7mAP higher than the traditional MobileV1-YOLOV4, 2.07 higher than MobileV3-YOLOV4. And MobileV1-YOLOV4 works better than MobileV3-YOLOV4. This shows that it is not that deeper networks can achieve better results. Compared with the mAP value of Faster-RCNN of 41.12, it can be concluded that in the case of a small data set, a deeper network may not be able to extract network features better.

2) Compared with the traditional YOLOV4 and Faster-RCNN, the GCB-Net proposed in this paper can better identify six defects (both obtained higher category AP values). After joining CBAM, compared with the Ghost-YOLOV4 network without joining, it can increase by 2.83 percentage points, respectively. This shows the effectiveness of adding CBAM to the enhanced feature extraction network, which will enhance the extraction of useful feature information and achieve better detection results.

3) Inconsistent detection difficulty across six different defects. The Cracking is the lowest, and the highest detection AP is only 0.59. The detection accuracy of Scratches can reach up to 0.97AP. The Scratch area is larger, and the pixels are clearer and easier to identify.

C) Compared with adding SE channel attention module for different attention:

Table 5 Comparison between models applying different attention.

Method	mAP	crazing	inclusion	patches	Pitted surface	Rolled-in scale	scratches

Ghost-3SE-YOLOV4	81.10	0.57	0.83	0.91	0.90	0.71	0.95
GCB-Net (our method)	82.78	0.59	0.86	0.93	0.92	0.70	0.97

It can be seen from Table 5 that adding attention will effectively improve the detection accuracy, especially for defect Crazing, which will significantly improve, and the effect of improving attention is also different. Among them, the network model with CBAM has a significant improvement, which is 2.83mAP higher than Ghost-YOLOV4.
D) Since the machine performance greatly influences the fps value, this paper tests on a 1080ti machine and detects the pictures through the model. The number of detections is 100 times, and then the average is taken. The comparison of parameters and model detection speed Frames Per Second (fps) is shown in Table 6 below:

Table 6 Model parameters and detection rate comparison.

Method	Parameters	fps
YOLOV4	64363101	57.21
MobileV1-YOLOV4	12692029	105.25
Ghost-YOLOV4	11428545	72.43
Ghost-3SE-YOLOV4	11433473	67.36
GCB-Net (our method)	11448711	66.14

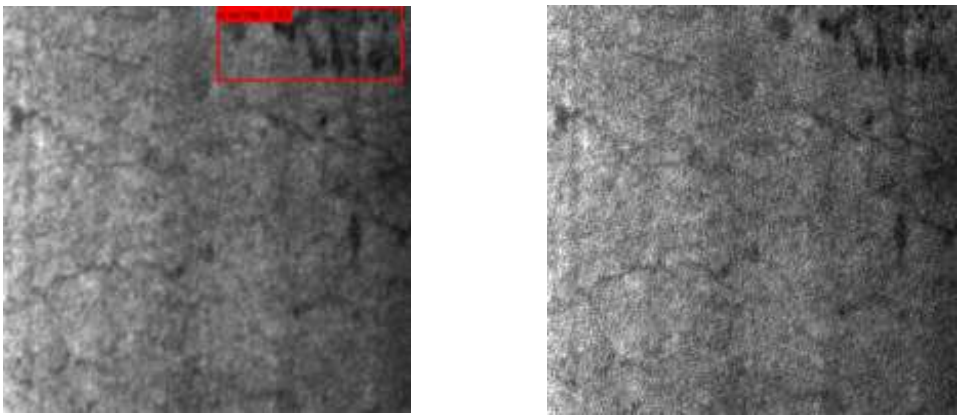
The following three conclusions can be drawn from the above table 6:

1) The detection rate of introducing Mobilenet > the detection rate of introducing Ghost > the detection rate of YOLOV4. The traditional YOLOV4 network is deeper, so the execution speed is slower.

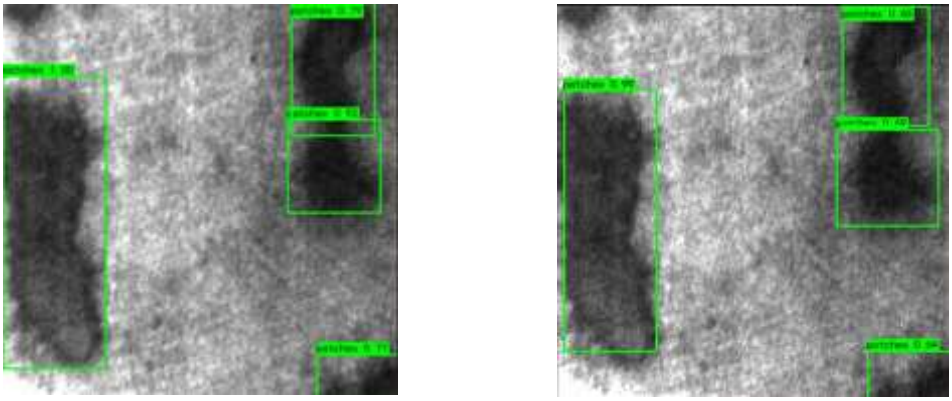
2) After the attention module is introduced, the number of parameters will increase accordingly, and the rate will decrease by 5-7 fps.

3) The execution rate of GCB-Net proposed in this paper is not comparable to Ghost-3SE-YOLOV4, but as shown in Table 5, mAP is increased by 1.68% and 1.22fps decrease speed. This indicates that the GCB-Net network can obtain higher detection accuracy under slightly reducing the speed, and the fps of 66.14 can meet industrial needs.

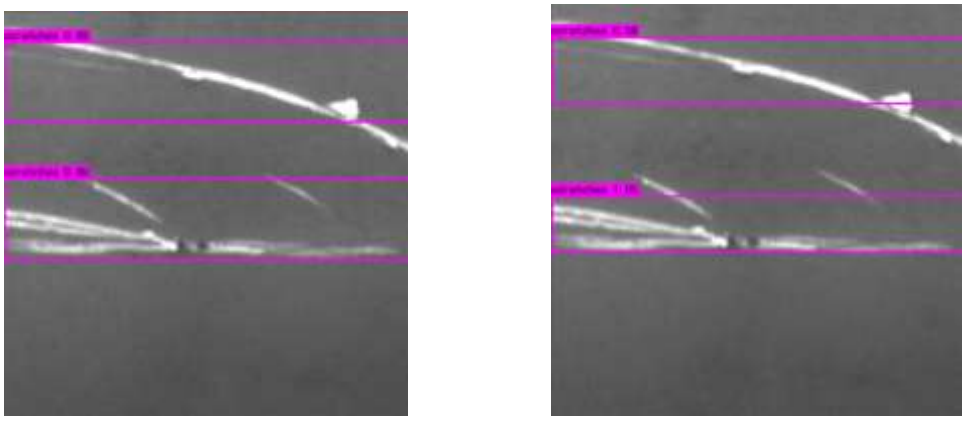
E) Randomly extract the defect pictures Crazing_28, Patches_128, Scratches_294, and compare the detection performance of the two models by comparing GCB-Net and traditional YOLOV4, As shown in Figure 14 below:



(a) Crazing_28 Defect image



(b) Patches_128 Defect image



(c) Scratches_294 Defect image

Fig.14 GCB-Net and YOLOV4 defect image comparison. By comparing images Crazing_28, Patches_128, and Scratches_294.

By observing the above three sets of defect detection pictures, the left side is the detection result of GCB-Net, and the right side is the detection result of YOLOV4. The following conclusions can be drawn:

1) Due to the poor overall fitting effect of the Crazing defect category, YOLOV4 cannot identify defects in Crazing_28 images. In contrast, GCB-Net can identify them more accurately, with a recognition score of 51.

2) In the Patches_128 image, the recognition accuracy of GCB-Net is higher than that of YOLOV4 due to defects in multiple positions. Like the defect in the upper right corner, GCB-Net scores 79, while YOLOV4 scores 65.

3) The overall fitting performance of the Scratches category in the data set is better, which is also due to the prominent defect location and obvious features, which can better extract useful information. It can be seen from the third set of comparisons that the recognition accuracy of YOLOV4 is slightly higher than that of GCB-Net due to the deep network, but both of them accurately identify defects.

5 Discussion

This paper builds an end-to-end defect detection network by introducing Ghost Module, and CBAM compares it with traditional YOLOV4, lightweight network MobileNet, etc., and achieves good experimental results. However, due to the small number of samples, although four online data enhancement and Mosaic data enhancement methods are used in this paper, the data set is still not very large, especially for the defect Crazing. Only the AP effect of 0.59 is obtained, indicating that the fitting effect is not good enough. Next, targeted data generation enhancement methods should be adopted for the industrial defect dataset to obtain better detection performance.

6 Conclusion

The contributions of this paper are mainly in the following three aspects:

1) The Ghost Module was innovatively introduced to replace the CSPNet of the original YOLOV4, and it was found that this can effectively reduce the parameter amount of YOLOV4 to improve the detection rate.

2) On the three effective feature layers extracted from the original YOLOV4 backbone network, CBAM is innovatively introduced to enhance the attention information, which can effectively improve the detection ability of the defect network model.

3) By comparing the GCB-Net proposed in this paper with the original YOLOV4, Faster-RCNN, MobileNet networks, AP, mAP, Precision, Recall, F1-score, FPS indicators are used for evaluation. It is found that GCB-Net can finally reach the detection accuracy and rate of 82.78mAP and 66.14fps, which has good industrial application value.

References

[1] C. Saunders, M.O. Stitson, J. Weston, R. Holloway, L. Bottou, B. Scholkopf, A.J.C.e. Smola, Support Vector Machine, 1(4) (2002) 1-28.

[2] T. Lindeberg, Scale Invariant Feature Transform, Scholarpedia2012.

[3] T.P.J.T.I.J.o.A.M.T. Lin, Fast Defect Detection in Textured Surfaces Using 1D Gabor Filters, (2002).

[4] H. Zheng, L.X. Kong, S.J.J.o.M.P.T. Nahavandi, Automatic inspection of metallic surface defects using genetic algorithms, s 125–126 (2002) 427-433.
[5] H. Hu, Y. Liu, M. Liu, L.J.N. Nie, Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm, 181(Mar.12) (2016) 86-95.
[6] J. Xi, S. Lifeng, J. Hu, M.J.A.O. Li, Automated surface inspection for steel products using computer vision approach, 56(2) (2017) 184-.
[7] H.Y. Wang, J. Zhang, Y. Tian, H.Y. Chen, H.X. Sun, K.J.I.T.o.I.I. Liu, A Simple Guidance Template-Based Defect Detection Method for Strip Steel Surfaces, (2018) 1-1.
[8] Y. Gao, L. Gao, X. Li, X.J.R. Yan, C.I. Manufacturing, A semi-supervised convolutional neural network-based method for steel surface defect recognition, 61(Feb.) (2020) 101825.1-101825.8.
[9] D. He, K. Xu, P. Zhou, D.J.O. Zhou, L.i. Engineering, Surface defect classification of steels with a new semi-supervised learning method, 117(JUN.) (2019) 40-48.
[10] K. Song, H. Dong, Y.J.O. Yan], L.i. Engineering, Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network, (2019).
[11] Y. He, K. Song, Q. Meng, Y.J.I.T.o.I. Yan, Measurement, An End-to-end Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features, PP(99) (2019) 1-1.
[12] K. Song, Y.J.A.S.S. Yan, A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects, 285(21) (2013) 858-864.
[13] A. Bochkovskiy, C.Y. Wang, H. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, (2020).
[14] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, GhostNet: More Features From Cheap Operations, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[15] S. Woo, J. Park, J.Y. Lee, I.S.J.S. Kweon, Cham, CBAM: Convolutional Block Attention Module, (2018).
[16] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, (2015).
[17] 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).
[18] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Searching for MobileNetV3, (2019).
[19] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, Q.J.I.T.o.I.I. Meng, PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection, PP(99) 1-1.
[20] R. Hao, B. Lu, Y. Cheng, X. Li, B.J.J.o.I.M. Huang, A Steel Surface Defect Inspection Approach towards Smart Industrial Monitoring, (9) (2020).
[21] G. Song, K. Song, Y.J.I.T.o.I. Yan, Measurement, EDRNet: Encoder–Decoder Residual Network for Salient Object Detection of Strip Steel Surface Defects, PP(99) (2020) 1-1.
[22] B. Gsa, B. Ksa, B.J.O. Yya, L.i. Engineering, Saliency detection for strip steel surface defects using multiple constraints and improved texture features, 128.
[23] J.P. Yun, J.L. Sang, G. Koo, C. Shin, C.H.J.O.e. Park, Automatic defect inspection system for steel products with exhaustive dynamic encoding algorithm for searches, 58(2) (2019) 023107.1-023107.9.
[24] H. Fukui, T. Hirakawa, T. Yamashita, H. Fujiyoshi, Attention Branch Network: Learning of Attention Mechanism for Visual Explanation, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[25] M.T. Luong, H. Pham, C.D.J.C.e. Manning, Effective Approaches to Attention-based Neural Machine Translation, (2015).

[26] H. Jie, S. Li, S. Gang, S.J.I.T.o.P.A. Albanie, M. Intelligence, Squeeze-and-Excitation Networks, PP(99) (2017).
[27] Z. Laskar, J.J.S. Kannala, Cham, Context Aware Query Image Representation for Particular Object Retrieval, (2017).
[28] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image Captioning with Semantic Attention, IEEE Computer Society, 2016.
[29] S. Zhang, X. Xu, Y. Pang, J.J.N.P.L. Han, Multi-layer Attention Based CNN for Target-Dependent Sentiment Classification, (2019).
[30] J. Redmon, A.J.a.e.-p. Farhadi, YOLOv3: An Incremental Improvement, (2018).
[31] K. He, X. Zhang, S. Ren, J.J.I. Sun, Deep Residual Learning for Image Recognition, (2016).