Application of YOLOv5 Neural Network Based on Improved Attention Mechanism in Recognition of Thangka Image Defects

Yao Fan[1], Yubo Li[1*], Yingnan Shi[1] and Shuaishuai Wang[1]
[1]School of Information Engineering, Xizang Minzu University Xianyang 712082, China
[e-mail: 93884969@qq.com, 1156760305@qq.com, 17610613383@163.com, 991248918@qq.com]
*Corresponding author: Yubo Li

Received September 23, 2021; revised December 2, 2021; accepted January 15, 2022;
published January 31, 2022

Abstract

In response to problems such as insufficient extraction information, low detection accuracy, and frequent misdetection in the field of Thangka image defects, this paper proposes a YOLOv5 prediction algorithm fused with the attention mechanism. Firstly, the Backbone network is used for feature extraction, and the attention mechanism is fused to represent different features, so that the network can fully extract the texture and semantic features of the defect area. The extracted features are then weighted and fused, so as to reduce the loss of information. Next, the weighted fused features are transferred to the Neck network, the semantic features and texture features of different layers are fused by FPN, and the defect target is located more accurately by PAN. In the detection network, the CIOU loss function is used to replace the GIOU loss function to locate the image defect area quickly and accurately, generate the bounding box, and predict the defect category. The results show that compared with the original network, YOLOv5-SE and YOLOv5-CBAM achieve an improvement of 8.95% and 12.87% in detection accuracy respectively. The improved networks can identify the location and category of defects more accurately, and greatly improve the accuracy of defect detection of Thangka images.

Keywords: YOLOv5, Defect Detection, Thangka Image, Deep Learning, SE, CBAM

This work was supported by Natural Science Foundation of China (No.62062061), the Xizang Natural Science Foundation (XZ202001ZR0065G, XZ202101ZR0084G)

http://doi.org/10.3837/tiis.2022.01.014 **ISSN：1976-7277**

1. Introduction

Thangkas have been hailed as an "encyclopedia of Tibetan culture" since ancient times. Drawing on various cultures throughout the history, they have developed into a distinctive cultural and artistic form representing Tibetan Buddhism and the characteristics of the snowy plateau [1]. Because of the unique production process of Thangkas, they are usually fragile and difficult to preserve. Therefore, the protection of Thangkas is more of a work of restoration. At present, most of the restoration of intangible cultural heritage is completed manually, with a large number of experienced experts examining the defective cultural relics and delineating the damaged area more accurately to achieve better repair effect. However, this task sets extremely high requirements for the workers. First of all, they must possess adequate professional knowledge and solid hands-on ability. It is also necessary for them to have a broad understanding of history, culture, archaeology, fine arts, humanities, and other subjects. The inspection of cultural relics should be done with great care, and "secondary damage" should be avoided. These high requirements have led to a shortage of experts [2]. Therefore, using computer vision to protect cultural heritage has been an inevitable development trend. For effective conservation and restoration, more accurate detection of defective areas is important. Hence, it is necessary to propose a new method for the defect detection of cultural heritage.

There has been a certain amount of research on detecting defective areas in China. Zhao et al. [3] proposed a defect detection framework based only on positive sample training. GAN and autoencoder were used to reconstruct the defective negative sample image, and LBP was used to compare the positive sample and the defect sample to detect the defect area of the cloth. This method only needs positive samples. Mei et al. [4] proposed an autoencoder network that used convolutional denoising on multiple Gaussian pyramid levels. It was applied to fabric defect detection and integrated with the detection results of the corresponding resolution channel. The reconstructed residuals were synthesized at each resolution level to generate the defect area. Zhang et al. [5] used Faster R-CNN and YOLOv3 to detect the aluminum defect data set, and applied it to the field of industrial models. Chen [6] used the YOLOv3 algorithm to identify five kinds of defects on the aircraft surface. Xu et al. [7] used the improved Mask R-CNN algorithm to detect tunnel defects, and endowed it with a path-enhanced feature pyramid network (PAFPN) and an edge detection branch. Wang et al. [8] proposed an improved Generative Adversarial Network (IGAN) method to detect machining surface defects. In this method, the Otsu algorithm was used to determine the residual image threshold and repair it, and then the input image was compared with the repaired image to obtain the defect area. Cha et al. used Faster R-CNN to detect images of edges and steel structures; this was the first time that Faster R-CNN had been applied to industry detection [9]. Tabernik et al. [10] proposed a deep learning system based on segmentation for detecting and segmenting defective areas on metal surfaces. This method can achieve a good detection effect even with a small number of defect samples [11].

With the development of deep learning technology in the field of computer vision, fruitful research results regarding defect detection have been achieved in recent years [12-17]. This paper applies the YOLOv5 network to the defect detection of Thangka images. It is found that the YOLOv5 network could not learn the characteristics of the defect area well for images with complex background color, and the detection results have shortcomings such as missed detection, false detection, and low detection accuracy. In order to solve the above problems, we propose an improved YOLOv5 algorithm based on attention mechanism to detect the defect of Thangka images with complex background color. This method not only improves the ability of the network to extract the texture and semantic features of the defect area, but also enables the attention mechanism of the network fusion to play an effective role in detection.

In summary, the main contributions of our work can be described as follows：

(1) Since the original network cannot fully learn the characteristics of the defect area of the Thangka pictures with complex background color, the SE (Squeeze-and-Excitation) mechanism is added after the output of Backbone, thus improving the feature learning ability of the network.

(2) The CBAM (Convolutional Block Attention Mechanism) mechanism is introduced after the output of Backbone for more detailedly allocating and dealing with the defect area as well as fusing the features. The addition of CBAM to the Neck network further enhances the feature extraction and accuracy of the improved network.

(3) The GIoU (Generalized-IoU) loss function of the YOLOv5 network is replaced with the CIoU (Complete-IoU) loss function. As a result, even if the detection box and the ground truth box overlap, the position information of the defect can still be effectively obtained and the convergence be accelerated.

(4) While applying the algorithm to the defect detection of Thangka images, this paper improves its efficiency of detection and reduces the probability of loss caused by human factors. This provides a new technical route for defect detection of Thangka images and a novel idea for digital protection of cultural heritage.

2. Network Framework

2.1 Improved algorithm

According to the features of Thangka images and the defect area and combined with the characteristics of the YOLOv5 network, the present paper proposed the YOLOv5 algorithm based on the attention mechanism. It mainly focused on the overall architecture design and optimization of the network, the improvement of the loss function, and the comparison with other networks. The problems of low accuracy and poor effect of defect detection were effectively solved, and the defect area of thangka images was identified more completely. Thus, the detection of Thangka image defects was completed successfully.

2.1.1 Design and optimization of the overall network architecture

The overall framework of the network is composed of Anchors, Backbone, Neck, and Prediction. In the input module, Mosaic data enhancement (random scaling, cropping, and arrangement for stitching), adaptive anchor box calculation, and adaptive image scaling were employed. Focus structure and BCSP structure were used in the Backbone module. The structure of FPN+PAN was adopted in the Neck module. Finally, the loss function of GIoU [18] was adopted in the Prediction module and NMS non-maximum suppression [19] was used to detect and classify targets.

The overall architecture improved the design of the original network from two aspects. One was to import the SE module and optimize it by filtering the relationship between channels, so that the characteristics output by Backbone could be further optimized and purified. The second was to import the CBAM module and optimize the features output by Backbone using channel attention and then spatial attention. Such combination could increase the learning rate of the network, and then feature fusion was performed to reduce data loss. Next, the two modules were imported into the network architecture separately.

This paper set three $anchors$, and used heads with different scales to detect defect targets of different sizes. Specifically, $class$ is the number of target categories to be detected, p is the probability, s is the coordinate of the defect center and the length and width of the inspection box, and $anchors$ is the number of the inspection boxes.

$$Channels = (class + p + s) \cdot anchors \tag{1}$$

2.1.2 SE mechanism

SE mechanism has a simple concept framework. The addition of SE does not need to change the overall framework of the network; it only needs to be added to the Backbone network. The idea of SE is to learn the correlation between channels [20]. The overall framework is shown in **Fig. 1**, where SE is in the red box. SE mainly explores the relationship between channels. Attention operations could be performed on its dimensions to make the network focus on the channel feature with the largest amount of information, suppress those unimportant channel features, reduce the computational load and complexity of the network, realize the adaptive selection of channel features, and model the selected relationships. Though the operating speed was decreased to some extent because adding the modules increased the depth and complexity of the network, its accuracy was improved dramatically.

SE includes three operations: Squeeze, Excitation, and Reweight. The Squeeze operation was to compress the H×W×C input from Backbone through mean-pooling and max-pooling, make the output dimensions and the number of characteristic channels match with each other, and obtain the global receptive field. The Excitation operation generated weights for feature channels through parameters. The Reweight operation was to weight the weights generated by Excitation to previous features via multiplication and fuse them to complete the learning of features.

The improvement made by this paper was reflected by the SE optimization of the features output by the network Backbone module, the addition of the SE mechanism to the Backbone module, and the deeper optimization learning performed on the features output by Backbone. In so doing, the network could better learn the defect features of the Thangka images with complex background color.

The SE mechanism first performed a global mean-pooling on the feature map with the input dimension of H×W×C, so a 1×1×C global receptive field was attained. Then through a full connection layer to get the characteristic results via the Sigmoid activation function, the correlation between the channels was obtained and the model was constructed. The result achieved was used as a weight to multiply the input feature.

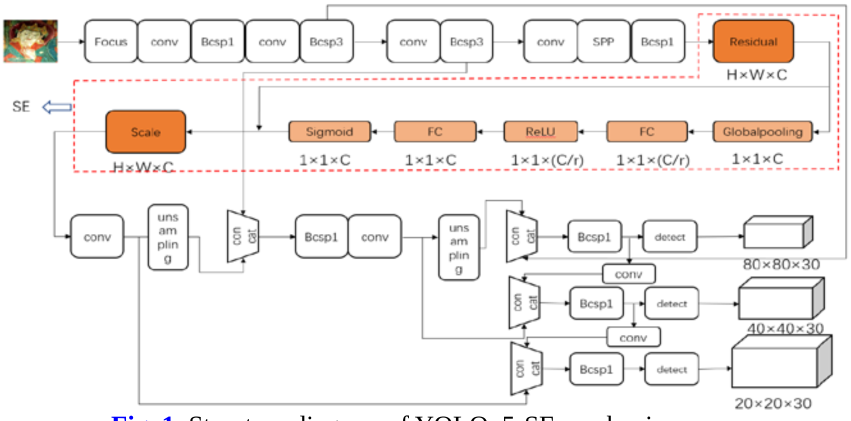


Fig. 1. Structure diagram of YOLOv5-SE mechanism

2.1.3 CBAM mechanism

CBAM (Convolutional Block Attention Module) is a typical type of attention mechanism. Its modules are enclosed in the red dashed box in **Fig. 2**. The core of the CBAM module includes the channel attention module in the green box and the spatial attention module in the blue box. A large number of experiments have proved that using channel attention first and then using spatial attention can achieve the best effect for network learning [21]. Using mean-pooling and max-pooling can effectively reduce the error rate, thereby obtaining a 1-2% improvement for the network and providing more detailed feature information. It is of great significance to the improvement of the network model, so this is currently the common combination method of CBAM.

Adding the CBAM mechanism to the lightweight model can greatly improve the performance of the network, though it will increase the network complexity and depth to some extent. Nevertheless, judging from the final test results, the CBAM approach is very effective given that it can achieve a massive increase in accuracy at the small cost of time.

For the CAM module in the CBAM mechanism, first the input H×W×C feature maps were subjected to mean-pooling and max-pooling respectively to obtain two 1×1×C feature maps, which were then successively sent to MLP. The number of neurons in the first layer of MLP is C/r, and the activation function is ReLU; the number of neurons in the second layer is C; the neural network of the two layers of MLP is universal. The features output by MLP were summed based on elementwise, and then the final channel attention feature was generated through the Sigmoid function. Finally, elementwise multiplication was performed between it and the input feature map to get the input features of Spatial attention.

For the SAM module in the CBAM mechanism, its input is the output of CAM. First, mean-pooling and max-pooling were performed to obtain two H×W×1 feature maps, which were then subjected to channel splicing. After 7×7 convolution, the dimensionality was reduced to H×W×1. Next, Spatial attention feature was generated through Sigmoid function, and finally it was multiplied by the input of the module to obtain the final generated feature [22]. The following formulas are the weight coefficient of the output of channel attention mechanism and spatial attention mechanism respectively.

$$M_{\mathrm{c}}(F)=\varsigma(MLP(MaxPool(F))+MLP(AvgPool(F))) \tag{2}$$

$$M_{x}(F)=\varsigma(f^{7\times7}((MaxPool(F);AvgPool(F)))) \tag{3}$$

Among them, ς is the Sigmoid operation, $MaxPool$ is the max-pooling, $AvgPool$ is the mean-pooling, $M_C(F)$ represents the weight coefficient of the output of channel attention mechanism, and $M_X(F)$ represents the weight coefficient of the output of spatial attention mechanism. Finally, the weight coefficient and the feature were multiplied together. Noteworthily, 7×7 represents the size of the convolution kernel.

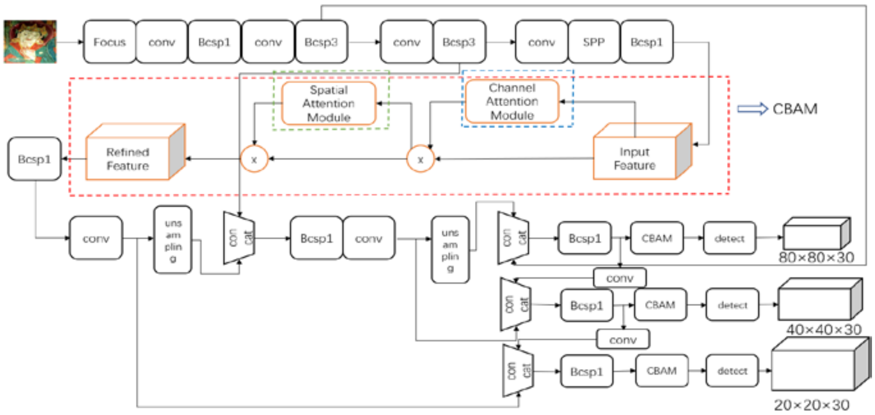


Fig. 2. Structure diagram of YOLOv5-CBAM mechanism

2.2 Loss function

Loss function is a tool for measuring the quality of a network's prediction result. This section elaborated the loss function of the YOLOv5 network. In addition, the loss function of the original network was improved, the improved loss function was described in detail, and each parameter of the loss function was explained and analyzed detailedly.

YOLOv5 uses the Cross-Entropy Loss Function to calculate the loss of the class probability of the sample and the confidence score of the target. Meanwhile, it uses $GIoU$ loss function as the loss of the bbox (bounding box). Compared with IoU [23] loss function, $GIoU$ increases the penalty for misdetection. The greater the detection error, the severer the penalty. In the process of training, relatively sound detection results could be obtained for prediction boxes of different sizes. But when the prediction box and the ground truth box overlap, the effect of $GIoU$ would be the same as that of IoU.

$$L = -\sum_{n=1}^{N} y^{(n)} \log x^{(n)} + (1 - y^{(i)}) \log(1 - x^{(n)}) \tag{4}$$

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} \tag{5}$$

$$GIoU = IoU - \frac{|D - (X \cup Y)|}{|D|} \tag{6}$$

$$L_{GIoU} = 1 - GIoU \tag{7}$$

The loss function of the YOLOv5 network in this paper modified in response to the above problems used $CIoU$. Different from $GIoU$ which calculates the intersection and union between the ground truth box and the prediction box, $CIoU$ calculates the Euclidean distance between the center points of the ground truth box and the prediction box, so $CIoU$ can solve the problems appeared while using $GIoU$. There are three important factors in the predicted bbox, which are the distance between the center points of the ground truth box and the prediction box, the overlap area, and the aspect ratio. When the prediction box and ground truth box overlap, $CIoU$, with a larger loss value than $GIoU$, can better describe the current position information. $CIoU$ considers both the overlap area of the prediction box and the ground truth box and the distance between their center points. The consistency in the aspect ratio of the bounding box is another important geometric factor, and $CIoU$ can normalize the distance between the two center points, thus accelerating the convergence of the network.

$$R_{CIoU} = \frac{\rho^2(X,Y)}{b^2} + \alpha \upsilon \tag{8}$$

$$\upsilon = \frac{4}{\pi^2}(\arctan\frac{Y_\omega}{Y_h} - \arctan\frac{X_\omega}{X_h})^2 \tag{9}$$

$$CIoU = 1 - IoU + R_{CIoU} \tag{10}$$

Among them, L is the Cross-Entropy Loss Function, X represents the probability of the predicted sample, Y represents the label, X is the prediction box, and Y is the ground truth box. As shown in **Fig. 3**, no matter whether the X box and the Y box intersect, the D box can contain X and Y boxes at the same time. $GIoU$ calculates the ratio of the area of D that is not covered by $X \cap Y$ to the area of D. Then, the ratio of the intersection of X and Y to their union minus the ratio obtained above can be obtained. Like IoU, $GIoU$ can also be used as a distance. When X and Y do not intersect, IoU is 0. The closer the ratio of $\frac{|D-(X \cup Y)|}{|D|}$ to 0, the closer the value of L_{GIoU} to 1. R_{CIoU} is its penalty term formula, ρ represents the Euclidean distance between the center points of the two prediction boxes of X and Y, b represents the diagonal distance of the smallest closure area that can contain both X and Y, α is the balance ratio parameter, and υ represents the similarity parameter that measures the length and width of the prediction box and the ground truth box.

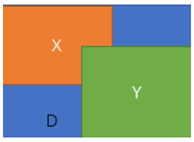


Fig. 3. The relationship between the rectangular boxes of X, Y, and D

3. Training Process

3.1 Training process of YOLOv5-SE

Fig. 4 shows the flow chart of YOLOv5-SE framework training and the convolution parameters and number of channels for each layer. First, in the Backbone module, the number of channels of the input image was expanded, and then the convolution operation was used to extract the shallow features of the input image. Four convolution layers were used, each containing a BN layer and an activation function ReLU. The features output from the Backbone passed through the SE module. The SE module first performed a global average pooling on the input 512×512×40 feature map to obtain a 1×1×40 global receptive field. Then the correlation between channels was obtained and the corresponding model was built through a full connection layer and using the Sigmoid activation function. The results obtained were used as weights to perform the elementwise operation with the input features, which enabled the network to learn the features of the defect area with complex background color more deeply. The feature representation became more distinguishable, thus improving the overall training effect of the network and enhancing the effect of the identification of defect area. Next, the extracted features were input to the Neck module which adopted the FPN+PAN combination. FPN fused the high-level semantic features through up sampling with the low-level texture features from top to bottom. Then the localization features of PAN network from bottom to top were fused, so as to realize the fusion of the parameters of different detection layers, thus strengthening the network's ability to fuse features. The fused features were up sampled and three feature maps with different dimensions were output. Finally, $CIoU$ was used as the loss function of the bounding box and NMS non-maximum suppression was employed to identify and locate the defect area of the input image more accurately.

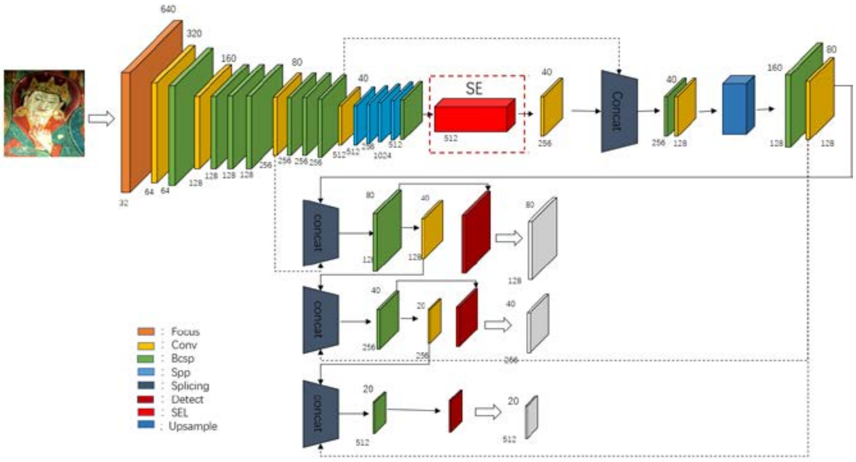


Fig. 4. Picture size and channel number of each layer of YOLOv5-SE

3.2 Training process of YOLOv5-CBAM

Fig. 5 shows the flow chart of YOLOv5-CBAM framework training and the convolution parameters and number of channels for each layer. First, in the Backbone module, the number of channels of the input image was expanded, and then the convolution operation was used to extract the shallow features of the input image. Four convolution layers were used. The features output by the Backbone module were input into the CBAM module, thus enabling the network to extract more distinguishable feature representation from the complex background color. The purpose of reducing information loss was achieved by feature fusion, and the features were then output to the next module. The CBAM module first passed through the channel attention module. Two feature maps were obtained after performing global average pooling and maximum pooling on the input features. Then elementwise operation was performed on the output features through the two-layer neural network of MLP. Another elementwise operation was performed between the generated features and the input feature through the Sigmoid activation function. Next, the obtained features were input into the spatial attention module for deeper learning. Global average pooling and maximum pooling were performed again on the input features, and then the obtained feature maps were subjected to channel splicing. Via 7×7 convolution operation, dimensionality reduction, and the Sigmoid activation function, the features were generated and subjected to the elementwise operation with the features generated by the channel attention module. The final generated features were obtained. After that, the extracted features were input to the Neck module, and the same FPN+PAN combination was used to process the features. However, unlike the previous process, at this point, before outputting the feature maps of different dimensions, the features of different dimensions were input into the channel attention module and the spatial attention module respectively for deeper learning and feature fusion, thus allowing the output feature maps with three different dimensions to carry more texture features and semantic features of defect areas. Finally, $CIoU$ was used as the loss function of the bounding box and NMS non-maximum suppression was employed to make its classification and positioning more accurate and improve the detection accuracy.

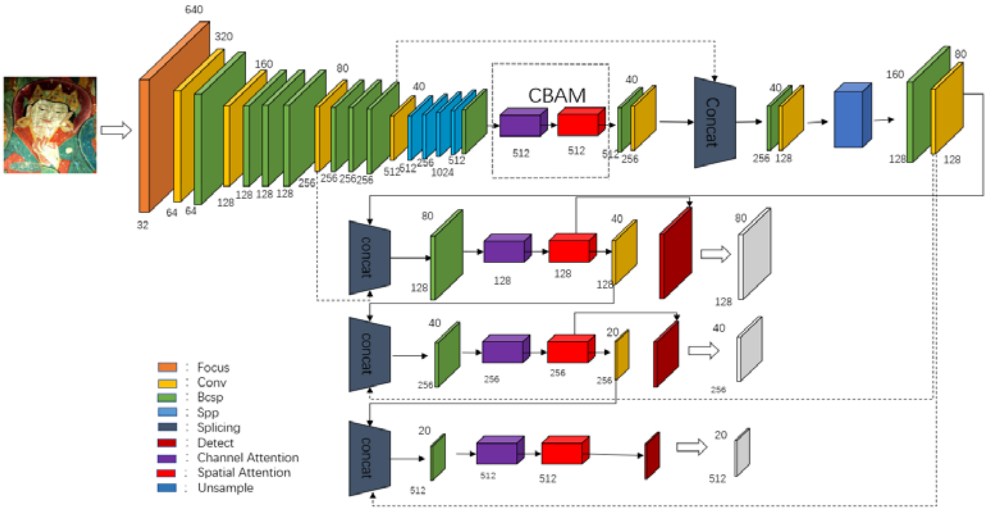


Fig. 5. Picture size and channel number of each layer of YOLOv5-CBAM

3.3 Training and testing process

This network process was mainly divided into two parts, namely the training of the defect detection network model and the testing of the defect data set. In the training process, we used 0.937 momentum and 0.0005 loss attenuation. The initial learning rate was set to 0.01, and the Batch_Size was set to 4. The network constantly updated the model parameters during the training and the loss function also decreased in the process of back propagation.

When the loss function reached the expected value or the convergence tended to equilibrium, the defect detection model training was completed and the trained model was saved. Then the images with defects were placed into the trained model for detection. The specific algorithm flow description is shown in **Table 1**.

Table 1. Algorithm flow description

Algorithm flow description	
Input：	Defective Thangka data set with tags.
Output：	Trained Thangka defect detection model.
Step 1：	Input the preprocessed Thangka image into the network.
Step 2：	Judge whether the number of iterations exceeds the set epoch. If yes, go to Step 7; if not, go to Step 3.
Step 3：	Randomly extract training set data, and convolve the input image.
Step 4：	Match the learned features with the input tags to classify the defect features.
Step 5：	Judge whether the training image defect learning is completed. If yes, go to Step 6; if not, go to Step 3.
Step 6：	Update the loss parameter value and the parameter of the defect features.
Step 7：	Save the trained defect model.
Step 8：	Put the test image into the trained network model to detect the defect area.
Step 9：	Perform feature matching and feature classification on the test image.
Step 10：	Calculate the confidence level of the test image, and visualize the training process with tensorboard.
Step 11：	Output the image after detection.
Step 12：	End the process.

4. Experimental Results and Analysis

4.1 Experiment preparation

The hardware platform built in this experiment was: Intel(R)Core (TM) I9-10900K CPU, 32G memory, NCIDIA GeForce RTX 2070 graphics card. The software environment was: CUDA version 11.0, CUDNN version 8.0, Windows 10 operating system. Python 3.7 and PyTorch 1.7.0 framework were adopted for data test, and the compiler pyCharm2020_1.2 _x64 was used.

Because of the peculiarity and scarcity of Thangka images, there is no unified data set so far, and the existing Thangka images are not only limited in number, but also affected by various factors such as different degrees of damage, poor availability, and low resolution. Consequently, image collection and processing has become an important part of this experiment. The data used in this paper came from the Thangka pictures taken in Tibet. Thangka images with defects were selected from the acquired data set of 5277 Thangka images. The data set was divided into the training set and test set according to the ratio of 8:2 to train and test the network. Due to the special data set of this experiment and the small number of initial data, the training effect was not very obvious. Therefore, the data set was expanded by the method of data enhancement, which increased not only the number of data sets, but also the diversity of the training data, thereby making the data training achieve better results.

This paper used the labeling tool of LabelImg to label the Thangka data sets and classify the defect area. There are five types of defect targets: fade, crack, dent, damage, and stain. Fade defect means that the pigment on the surface of the Thangka only falls off slightly without damaging the bottom plate; Crack defect indicates that there is a serious crack in the middle of the Thangka and it has seriously damaged the bottom plate; Dent defect means that the surface of the Thangka is concave due to external force, and it becomes uneven and incomplete; Damage defect indicates that not only the surface color of the Thangka falls off, but also its bottom plate is slightly damaged; Stain defect indicates that the surface of the Thangka is contaminated with soil, oil, rain stains, or other pigment that does not belong to the original image. The defect distribution of the test set is shown in **Table 2**. The left sample is listed as the defect sample type of the test set, and the right is the distribution of the defect sample in the test set.

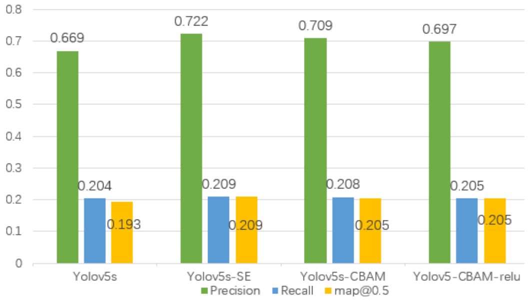
Table 2. Distribution of defect samples in the test set

Sample	All	Fade	Crack	Dent	Damage	Stain
N	576	210	187	30	107	42

4.2 Comparison test

4.2.1 Comparison of the effect of unexpanded data set

Fig. 6 and **Fig. 7** show the data analysis results obtained from the experiments of 3000 and 6000 iterations before data expansion.



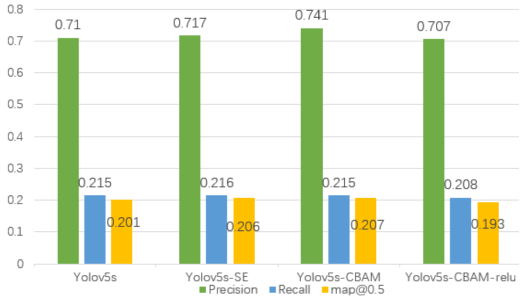


Fig. 6. Histogram of the effect of 3000 iterations **Fig. 7.** The histogram of the effect of 6000 iterations

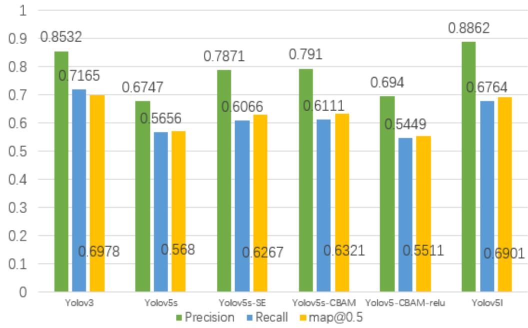
As observed from **Fig. 6**, when the unexpanded data set was used, YOLOv5-SE performed better when the epoch iterated 3000 times. YOLOv5-CBAM also exceeded the original model in terms of precision, recall rate, and @ 0.5. Nevertheless, when the Relu activation function was used in the experiment, despite the improved accuracy and recall rate, its time cost was increased and the precision and recall rate failed to achieve expected improvement compared with the improved algorithm. Therefore, it was removed in the follow-up comparison test. In terms of the time consumption of training, YOLOv5s as a lightweight network had a great speed advantage among the compared centralized algorithms. Compared with YOLOv5-CBAM, YOLOv5-SE increased the precision by 5.3% and 4%, and the recall rate by 0.5% and 0.4%.

Table 3. Time consumption of unexpanded data set training

Iterations	**YOLOv5s**	**YOLOv5-SE**	**YOLOv5-CBAM**	**YOLOv5-CBAM-Relu**
3000	2.378	**3.431**	**4.267**	4.301
6000	4.756	**7.94**	**9.125**	9.212

As shown in **Fig. 7**, when the epoch iterated 6000 times, the performance of YOLOv5-CBAM in terms of accuracy and recall rate surpassed that of YOLOv5-SE, and its @0.5 was also improved to a certain extent compared with the original model. However, with regard to time consumption, YOLOv5s still had great advantages. YOLOv5-CBAM training took nearly twice as long as the YOLOv5s training. Compared with YOLOv5-CBAM, YOLOv5-SE improved the precision by 0.7% and 3.1%, and the recall rate by 0.1% and 0%.

4.2.2 Comparison of the effect of expanded data set



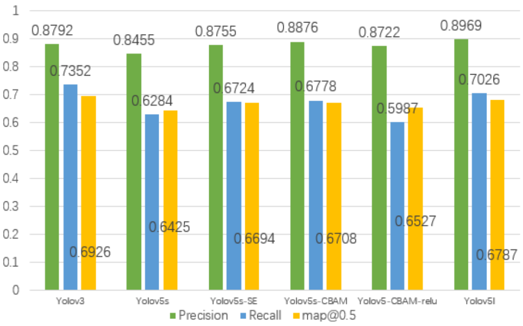


Fig. 8. Histogram of the effect of 1000 iterations **Fig. 9.** Histogram of the effect of 3000 iterations

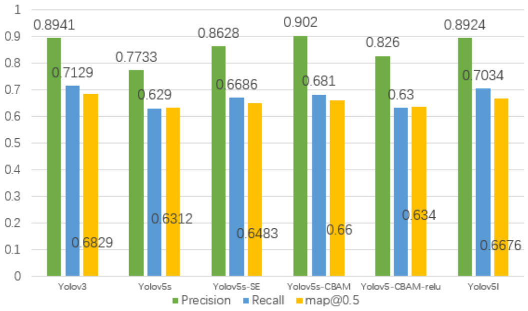


Fig. 10. Histogram of the effect of 6000 iterations

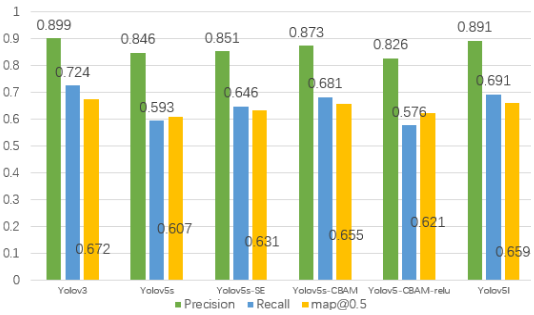


Fig. 11. Histogram of the effect of 10000 iterations

Table 4. Time consumption and FPS of expanded data set training

	Iterations	**YOLO v3**	**YOLO v5s**	**YOLO v5s-SE**	**YOLOv5-CBAM**	**CBAM-ReLU**	**YOLO v5l**
Time(h)	1000	8.74	1.2105	1.567	1.756	1.76	4.38
	3000	26.21	3.6315	4.703	5.2685	5.273	13.14
	6000	52.403	7.263	9.406	10.537	10.546	26.28
	10000	82.4	12.105	15.249	16.958	44.3	82.4
FPS	1000	6.7	31.88	28.95	24.55	23.56	8.74
	3000	6.7	31.88	28.95	24.55	23.56	8.74
	6000	6.7	31.88	28.95	25.23	23.56	8.74
	10000	6.7	31.00	28.44	25.23	8.74	6.7

As can be observed from Fig. 8, after the data set was expanded, the precision and recall rate of the improved network and other networks were greatly increased, which fully shows that the expansion of the data set had a very important impact on the prediction of network. When the epoch iterated 1000 times, YOLOv5l significantly outperformed other network models in terms of the detection accuracy. While the prediction effect of YOLOv3 was second only to YOLOv5l, and its recall rate was much better than that of other networks. From the comparison of @0.5, the average precision of YOLOv3 and YOLOv5l was also slightly higher than that of the original model and the improved network model. However, in terms of training time, that of YOLOv3 network was equivalent to the sum of that of other networks; YOLOv5s and other improved models had an absolute advantage in this aspect. From the perspective of the number of processed picture frames, that of YOLOv5l was 8.74, which was only slightly better than YOLOv3.The detection precision, recall rate, and detection speed of YOLOv5-SE and YOLOv5-CBAM were relatively balanced. Compared with YOLOv5-CBAM, YOLOv5-SE improved the precision by 11.24% and 11.63%, and the recall rate by 4.1% and 4.55%.

Fig. 9 shows that when the epoch iterated 3000 times, the detection accuracy and recall rate of YOLOv5l were better, closely followed by YOLOv5-CBAM. The detection effect of other network models was not very different. Nonetheless, although the precision and recall rate of YOLOv3, YOLOv5-SE, and YOLOv5-CBAM were slightly inferior to those of YOLOv5l, their training speed was much higher than that of YOLOv5l. Specifically, the speed of YOLOv5s was only a quarter of that of YOLOv5l, and one-eighth of that of YOLOv3. In terms of FPS, YOLOv5s was far faster than YOLOv3 and YOLOv5l. Compared with YOLOv5-CBAM, YOLOv5-SE improved the precision by 3.0% and 4.21%, and the recall rate by 4.4% and 4.94%.

According to **Fig. 10**, when the epoch iterated 6000 times, YOLOv5-CBAM had the best detection precision and recall rate. Specifically, its precision was more than 90%, which was the most prominent in the comparison experiment. Its @0.5 data also far outperformed that of others. In contrast, the precision of YOLOv5s was much inferior to that of other models. From the training time and FPS, YOLOv5s and the improved model still had much greater advantages than YOLOv3 and YOLOv5l. Compared with YOLOv5-CBAM, YOLOv5-SE increased the precision by 8.95% and 12.87%, and the recall rate by 3.96% and 5.2%.

Fig. 11 reveals that when the epoch iterated 10,000 times, the detection precision and recall rate of YOLOv5l and YOLOv3 were relatively good, and the data of @0.5 also had a great advantage in the comparison experiment. But the shortcomings were also obvious. Regarding the detection speed and FPS data, the time cost of YOLOv3 training was too high, and its FPS value was also the lowest among the five groups of comparisons. While the time cost of YOLOv5l training was half that of YOLOv3, it was still too high, being 3.6 times that of YOLOv5s; and its FPS was less than one-third of that of YOLOv5s. However, YOLOv5-CBAM's detection precision, speed, and recall rate were relatively balanced. Although its training time cost was higher than that of the original model, its accuracy and recall rate were greatly improved compared with the original model. Compared with YOLOv5-CBAM, the precision of YOLOv5-SE was increased by 0.4% and 2.7%, and the recall rate by 5.3% and 6.7%.

The experimental results show that the in terms of the precision, recall rate, and @0.5, the two networks improved based on YOLOv5s greatly outperformed the original YOLOv5s model, thus capable of extracting features of defect Thangka images with complex background more accurately. However, the addition of SE and CBAM mechanisms increased the depth of the network models, which sacrificed a certain speed advantage to improve the effect of detecting defects. This made the training time cost and FPS inferior to those YOLOv5s, but their precision and recall rate were close to those of YOLOv3 and YOLOv5l. On the other hand, their training time cost and FPS were greatly improved compared to YOLOv3 and YOLOv5l. The improved network models had the speed of YOLOv5s and also improved the defect detection accuracy and recall rate.

4.3 Comparison of the detection effect concerning the five types of defects

Table 5 below displays the comparative experimental data concerning the detection of the five types of defects, and the values in bold type denote the data of the improved networks.

Table 5. Detection data concerning Fade defect (Defect 0)

Iterations	Fade	P	R	map@0.5	map@0.5:0.95
6000	YOLOv3	0.857	0.895	0.844	0.653
	YOLOv5s	0.812	0.881	0.859	0.539
	YOLOv5-SE	**0.828**	**0.900**	**0.846**	**0.590**
	YOLOv5-CBAM	**0.858**	**0.890**	**0.850**	**0.588**
10000	YOLOv5s	0.834	0.876	0.816	0.449
	YOLOv5-SE	**0.831**	**0.881**	**0.830**	**0.514**
	YOLOv5-CBAM	**0.856**	**0.890**	**0.839**	**0.532**
	YOLOv5l	0.861	0.914	0.855	0.589

From **Table 5** we could see that for the comparison of the detection precision concerning the Fade defect, YOLOv5l had the best effect, followed by YOLOv5-CBAM and then YOLOv3. In terms of the recall rate, YOLOv5l performed the best, successively followed by YOLOv5-SE and YOLOv3. But the processing speed of YOLOv5-CBAM was 24.55 frames per second, much higher than that of YOLOv5l (8.74 frames per second) and YOLOv3 (6.7 frames per second).

Table 6. Detection data concerning Crack defect (Defect 1)

Iterations	**Crack**	**P**	**R**	**map@0.5**	**map@0.5:0.95**
6000	YOLOv3	0.785	0.850	0.720	0.504
	YOLOv5s	0.732	0.556	0.547	0.239
	YOLOv5-SE	**0.789**	**0.663**	**0.662**	**0.317**
	YOLOv5-CBAM	**0.780**	**0.610**	**0.606**	**0.324**
10000	YOLOv5s	0.754	0.444	0.515	0.183
	YOLOv5-SE	**0.748**	**0.588**	**0.554**	**0.237**
	YOLOv5-CBAM	**0.755**	**0.594**	**0.585**	**0.276**
	YOLOv5l	0.737	0.751	0.630	0.326

Table 6 illustrates that concerning the Crack defect, YOLOv5-SE achieved the best precision, followed by YOLOv3 and then YOLOv5-CBAM. With regard to the recall rate, the top three best-performed models were YOLOv3, YOLOv5l, and YOLOv5-SE successively. In addition, YOLOv5-SE processed 28.95 frames per second; YOLOv5-CBAM processed 24.55 frames per second, much better than YOLOv3 (6.7 frames per second).

Table 7. Detection data concerning Dent defect (Defect 2)

Iterations	**Dent**	**P**	**R**	**map@0.5**	**map@0.5:0.95**
6000	YOLOv3	0.930	0.367	0.365	0.286
	YOLOv5s	0.693	0.267	0.335	0.167
	YOLOv5-SE	**0.880**	**0.333**	**0.349**	**0.232**
	YOLOv5-CBAM	**0.950**	**0.364**	**0.365**	**0.235**
10000	YOLOv5s	0.858	0.267	0.324	0.143
	YOLOv5-SE	**0.886**	**0.333**	**0.340**	**0.168**
	YOLOv5-CBAM	**0.934**	**0.364**	**0.254**	**0.192**
	YOLOv5l	0.948	0.361	0.365	0.204

As can be seen from **Table 7**, the precision of YOLOv5-CBAM for the Dent defect was the highest, followed by YOLOv5l. However, the recall rate and precision for this type of defect were not high in general. The reason is that the data sample of this type of defect accounted for a too small percentage of the overall defect sample. The comparison test shows that the recall rate of YOLOv3 was the highest, followed by YOLOv5-CBAM and then YOLOv5l. The processing speed of YOLOv5-CBAM was 24.55 frames per second, much higher than that of YOLOv5l (8.74 frames per second).

Table 8. Detection data concerning Damage defect (Defect 3)

Iterations	Damaged	P	R	map@0.5	map@0.5:0.95
6000	YOLOv3	0.903	0.860	0.892	0.696
	YOLOv5s	0.869	0.822	0.829	0.501
	YOLOv5-SE	**0.908**	**0.828**	**0.835**	**0.572**
	YOLOv5-CBAM	**0.943**	**0.822**	**0.875**	**0.561**
10000	YOLOv5s	0.879	0.757	0.801	0.419
	YOLOv5-SE	**0.925**	**0.811**	**0.840**	**0.517**
	YOLOv5-CBAM	**0.923**	**0.832**	**0.865**	**0.519**
	YOLOv5l	0.931	0.813	0.854	0.578

As can be observed from **Table 8**, YOLOv5-CBAM had the best precision concerning the Damage defect, followed by YOLOv5-SE and then YOLOv3. For the recall rate, YOLOv3 performed the best, followed by YOLOv5-CBAM and then YOLOv5-**SE**. However, YOLOv5-SE processed 28.95 frames per second; YOLOv5-CBAM processed 24.55 frames per second, much better than YOLOv3 (6.7 frames per second).

Table 9. Detection data concerning Stain defect (Defect 4)

Iterations	Stain	P	R	map@0.5	map@0.5:0.95
6000	YOLOv3	0.926	0.593	0.593	0.535
	YOLOv5s	0.760	0.619	0.589	0.425
	YOLOv5-SE	**0.908**	**0.619**	**0.589**	**0.510**
	YOLOv5-CBAM	**0.928**	**0.619**	**0.606**	**0.487**
10000	YOLOv5s	0.906	0.619	0.577	0.441
	YOLOv5-SE	**0.861**	**0.619**	**0.591**	**0.434**
	YOLOv5-CBAM	**0.890**	**0.619**	**0.598**	**0.469**
	YOLOv5l	0.925	0.619	0.592	0.473

It can be seen from **Table 9** that YOLOv5-CBAM achieved the best precision for the Stain defect, followed by YOLOv3 and then YOLOv5l. Regarding the recall rate, YOLOv3 was the lowest, while that of the rest of the networks were the same. However, the processing speed of YOLOv5-CBAM was 24.55 frames per second, much higher than that of YOLOv5l (8.74 frames per second) and YOLOv3 (6.7 frames per second).

According to the above comparison tests, from the perspective of detection precision, YOLOv5-CBAM performed best on Defects 2, 3, and 4, YOLOv5-SE performed best on Defect 1, and YOLOv5l performed best on Defect 0. The precision of the improved networks was comparable to that of YOLOv5l and YOLOv3. From the perspective of detection speed and training time cost, compared with YOLOv5, YOLOv5-SE**,** and YOLOv5-CBAM, YOLOv3 and YOLOv5l greatly increased the detection speed and deduced the training time. Therefore, the improved networks achieved a relative balance in terms of precision and speed.

4.4 Comparison of the visual effect

4.4.1 Comparison of the visual effect of different bbox loss functions

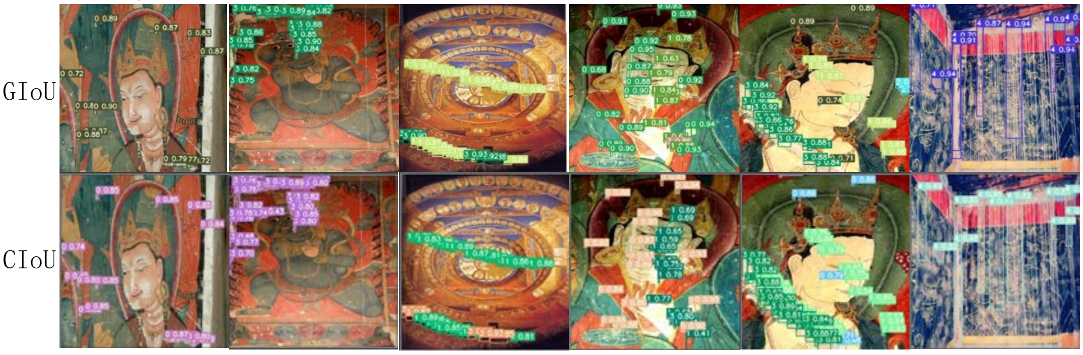


Fig. 12. Comparison of images of different bbox loss functions

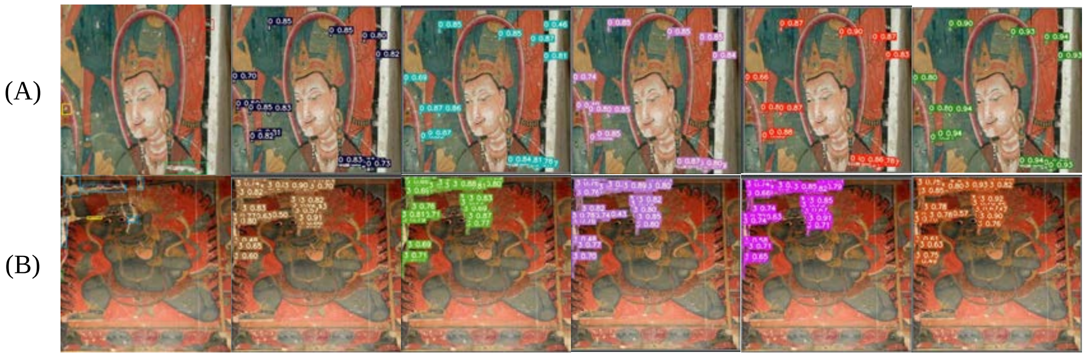
According to **Fig. 12**, the detection effect of YOLOv5-CBAM using GIoU as the bbox loss function was not good, and it was highly prone to miss detection and false detection. As shown in the fifth figure of GIoU, the yellow box area on the image suffered from severe loss of defect features, and the corresponding defect area was not detected. By comparison, CIoU detection not only greatly reduced the missed detection area, but also had a very low probability of false detection. The yellow defect area of several other images reflected miss detection to varied degrees. According to **Table 10**, when iterating 6000 times, the detection precision and recall rate of CIoU were slightly improved compared with those of GIoU.

Table 10. Detection effect of GIoU and CIoU after 6000 iterations

	Precision	Recall	map@0.5	map@0.5:0.95
CBAM-GIoU	0.862	0.656	0.647	0.427
CBAM-CIoU	**0.902**	**0.681**	**0.660**	**0.439**

4.4.2 Comparison of the visual effect of different defect detection algorithms

Fig. 13 shows an example of defect detection results of Thangka data with the same defects using different algorithms. We set the detection anchor point frame score threshold to 0.4; when the score of the anchor point frame was greater than 0.4, we output the sample as a positive sample; when it was less than 0.4, the sample was classified as a negative sample. Consequently, a large number of negative sample area proposals could be omitted, thereby saving time and cost. The following is the comparison result of the experimental images.



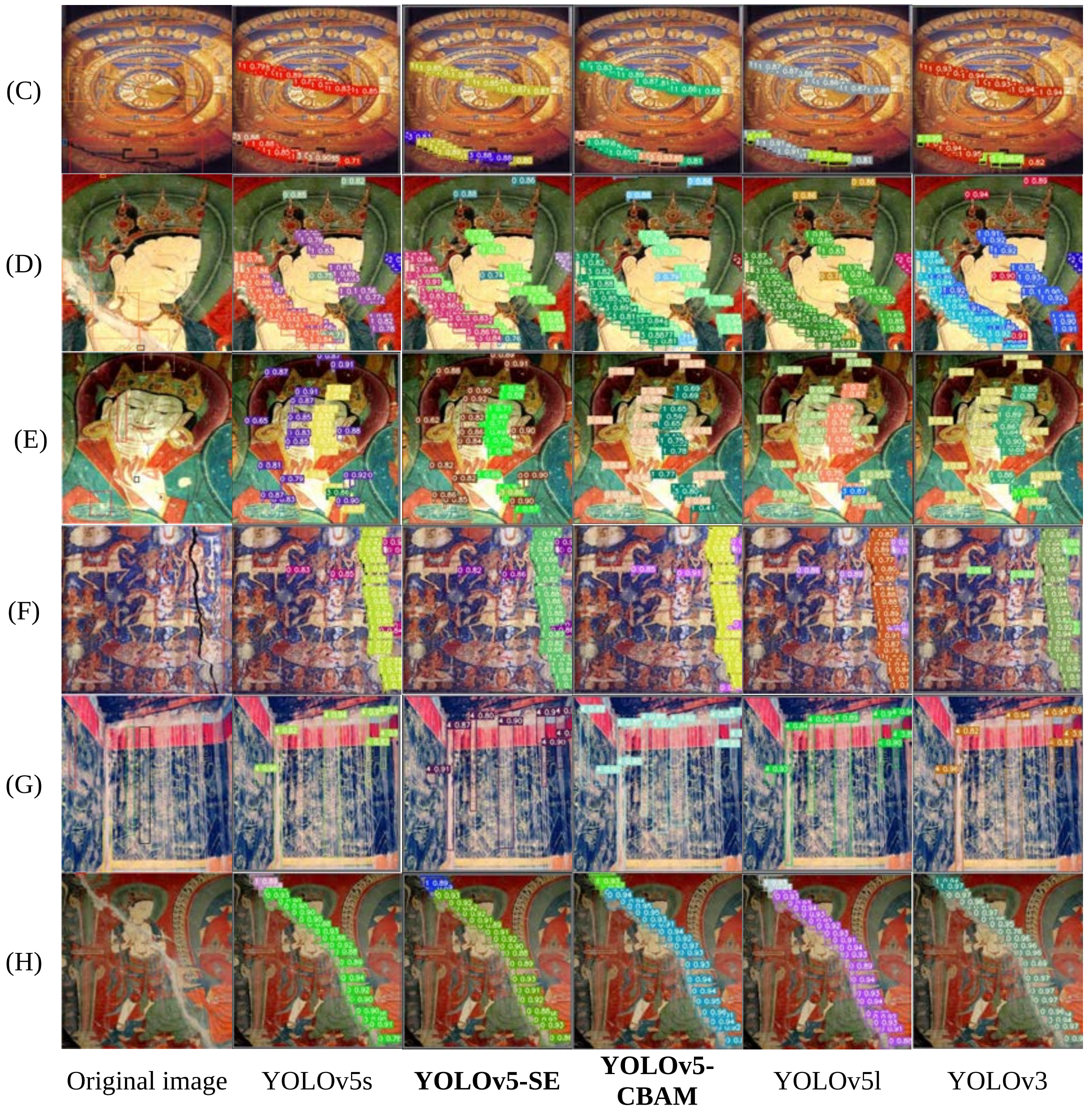


Fig. 13. Comparison of different algorithms for defective images

For Defect 0 marked in the red area of the original image of Group A, only YOLOv5-SE detected the defect, the *IoU* of which was 0.46; for Defect 0 in the yellow area, YOLOv5s and YOLOv5-CBAM succeeded in the detection, while YOLOv5-SE, YOLOv5l, and YOLOv3 failed; YOLOv5l and YOLOv3 did not detect Defect 0 in the red area and the yellow area accurately.

In the green marked area of the original image of group B, YOLOv5s failed to recognize Defect 3; Defect 3 in the red area was not recognized by the YOLOv5s and YOLOv5-SE models; Defect 3 in the yellow area was recognized by YOLOv5-SE, but not by YOLOv5l and YOLOv5l; in the blue area, all models identified the defect, and YOLOv5-CBAM achieved the best performance.

Only YOLOv3 did not recognize Defect 3 in the blue box in group C; at the same time, YOLOv5l misrecognized Defect 3 in the blue box as Defect 1.

In group D, only YOLOv5-SE and YOLOv5-CBAM accurately detected Defect 0 in the yellow box, indicating that when the attention mechanism module was added, the model could better learn the characteristics of the target area, thereby effectively identifying the defect area. Defect 3 in the black box was only accurately detected by YOLOv5l; Defects 1 and 3 in the orange box were also effectively identified by the five types of networks.

Defect 3 in the orange box in the original picture of Group E was relatively obvious, so all the network models recognized it; and for Defect 1 in the black box, YOLOv5s, YOLOv5-SE, and YOLOv5-CBAM all failed to recognize.

Only YOLOv5-CBAM accurately detected Defect 0 in the orange-yellow area of the original image of Group F. Compared with other models, YOLOv5-CBAM was able to perform detailed detection due to the addition of the CBAM module, which further illustrated that the improved model greatly enhanced the network.

Since the image in Group G was the most damaged, the detection can better reflect the effect of the models. For Defect 4 in the orange box and the black box, only YOLOv5-CBAM accurately detected it; for Defect 4 in the yellow box, YOLOv5-SE and YOLOv5l failed to completely detected it, missing the lower part of the defect; Defect 4 in the green box of the original image was not detected by YOLOv5s and YOLOv3.

The picture in Group H suffered from a continuous defect. For Defect 1 in the red box in the original picture, only YOLOv5-CBAM recognized it, while the other parts of the defect were completely recognized by all models.

Judging from the overall test results, the network models with SE and CBAM modules showed better learning ability, though sacrificing the speed and time to some extent. In specific, the networks with CBAM module adopted the combination of using channel attention first and then spatial attention, thus capable of learning more fully the small features of the defect area and detecting the characteristics of defects under the conditions of complex background. They had obvious advantages in the comparison of Groups C, E, F, G, and H. Furthermore, compared with the original YOLOv5l and YOLOv3 models, no misdetection occurred in the two improved network models, reflecting their higher reliability. The two improved network models also had advantages in terms of speed and training time compared with YOLOv3 and YOLOv5l. Compared with the original YOLOv5s network, although the improved networks' detection speed was slightly slower and training time was slightly higher, their accuracy and recall rate were better. In general, the improved networks had their own outstanding features and advantages in the comparison test. They could learn the advantages of other networks and overcome their shortcomings in the meanwhile, thus realizing excellent performance in terms of both the speed and precision.

4.5 Comparison of model size

Table 11. Comparison of the model size when the epoch iterated 10000 times

Network model	YOLOv3	YOLOv5s	**YOLOv5-SE**	**YOLOv5-CBAM**	YOLOv5l
Model size（MB）	117.75	13.76	**14.32**	**16.19**	89.44

From **Table 11** we could see that when the epoch iterated 10,000 times, the model trained by the YOlOv3 network was of the largest size, reaching 117.75 MB, while the model trained by the YOLOv5l network was also as large as 89.44 MB. In contrast, the model sizes of YOLOv5s, YOLOv5-SE, and YOLOv5-CBAM were only a dozen of MB, with the largest being 16.19 MB for YOLOv5-CBAM network model. These numbers were much smaller than the size of the other two network models, showing their huge storage space advantage compared to the other two models.

5. Conclusion and Future Work

Aiming at resolving the problems of difficult feature extraction and low detection accuracy concerning defective Thangka images with complex background color, this paper has proposed an improved YOLOv5 model based on attention mechanism to detect the defect area in Thangka images. The model integrated SE and CBAM mechanisms respectively, and thus obtained two improved algorithms, YOLOv5-SE and YOLOv5-CBAM, which have effectively solved the problems encountered in Thangka image defect detection.

The experimental results have shown that the improved defect detection models can accurately extract the features of the defect area from images with complex background color and detect multiple defects simultaneously. They have achieved improved detection accuracy and recall rate without greatly increasing the time cost, and their detection speed has been only slightly slower than that of YOLOv5s. However, compared with the speed of YOLOv5l and YOLOv3, the improved networks have had many advantages, and their precision has been far from inferior to that of other models. YOLOv5-SE has achieved an improvement of 8.95% in detection accuracy, and YOLOv5-CBAM an improvement of 12.87%. The improvement in accuracy of YOLOv5-CBAM has been more obvious than that of YOLOv5-SE. In general, the real-time performance of both the improved networks has been effectively enhanced, thereby capable of effectively solving the problems regarding defect extraction of Thangka images.

References

[1] L. Li, "Tibetan Buddhist Thangka Art," *Journal of Kangding Nationaly Teachers College*, no. 01, pp. 13-16, 2008. Article (CrossRef Link).

[2] L. Jie, "Research and implementation of Thang-ga image inpainting system based on deep learning," *Ningxia University*, 2020. Article (CrossRef Link).

[3] Zhao Z., Li B., Dong R., Zhao P, "A Surface Defect Detection Method Based on Positive Samples," in *Proc. of Pacific Rim International Conference on Artificial Intelligence (PRICAI 2018)*, Nanjing, China, vol. 11013, pp. 473-481, 2018. Article (CrossRef Link).

[4] M. Shuang, W. Yudan, and W. Guojun, "Automatic Fabric Defect Detection with a Multi-Scale Convolutional Denoising Autoencoder Network Model," *Journal of Sensors*, Basel, Switzerland, vol. 18, no. 4, 2018. Article (CrossRef Link).

[5] Z. Xu and H. Dingjiang, "Defect detection on aluminum surfaces based on deep learning," *Journal of East China Normal University (Natural Science Edition)*, no. 06, pp. 105-114, 2020. Article (CrossRef Link).

[6] C. Conghan, "Application of deep neural network based on YOLOv3 in aircraft surface defect recognition, " *Civil Aviation Flight University of China*, vol. 10, 2020. Article (CrossRef Link).

[7] X. Yingying, L. Dawei, X. Qian, W. Qiaoyun, and W. Jun, "Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN," *Journal of Measurement*, vol. 178, 2021.Article (CrossRef Link).

[8] W. Qingxia, Y. Ronghui, W. Chongjun, and L. Yao, "An effective defect detection method based on improved Generative Adversarial Networks (iGAN) for machined surfaces," *Journal of Journal of Manufacturing Processes*, vol. 65, no. 5, pp. 373-381, 2021. Article (CrossRef Link).

[9] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous Structural Visual Inspection Using Region‐Based Deep Learning for Detecting Multiple Damage Types," *Special Issue: Health Monitoring of Structures*, vol. 33, no. 9, pp. 731-747, 2018. Article (CrossRef Link).

[10] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759-776, 2020. Article (CrossRef Link).

[11] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386-397, 2020. Article (CrossRef Link).

[12] Wang, T., Chen, Y., Qiao, M. et al., "A fast and robust convolutional neural network-based defect detection model in product quality control," *Int J Adv Manuf Technol*, vol. 94, pp. 3465–3471, 2018.Article (CrossRef Link).

[13] J. Chen, Z. Liu, H. Wang, A. Nunez, Z. Han, "Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 2, pp. 257-269, 2018. Article (CrossRef Link).

[14] S. Mei, H. Yang, Z. Yin, "An Unsupervised-Learning-Based Approach for Automated Defect Inspection on Textured Surfaces," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1266-1277, 2018. Article (CrossRef Link).

[15] C. Feng, M. Y. Liu, C. C. Kao, and T. Y. Lee, "Deep Active Learning for Civil Infrastructure Defect Detection and Classification," in *Proc. of ASCE International Workshop on Computing in Civil Engineering*, 2017. Article (CrossRef Link).

[16] Lei et al., "Scale insensitive and focus driven mobile screen defect detection in industry," *Neurocomputing*, vol. 294, pp. 72-81, 2018. Article (CrossRef Link).

[17] D. Tabernik, S. Ela, J. Skvar, and D. J. J. o. I. M. Skoaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, pp. 759–776, 2020. Article (CrossRef Link).

[18] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, USA, pp. 658-666, 2019. Article (CrossRef Link).

[19] D. Oro, C. Fernandez, X. Martorell, and J. Hernando, "Work-efficient parallel non-maximum suppression for embedded GPU architectures," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, pp. 1026-1030, 2016. Article (CrossRef Link).

[20] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011-2023, 1 Aug. 2020. Article (CrossRef Link).

[21] Woo S., Park J., Lee JY., Kweon I.S, "CBAM: Convolutional Block Attention Module," in *Proc. of the European Conference On Computer Vision (ECCV 2018)*, Munich, Germany, vol. 11211, pp. 3-19, 2018. Article (CrossRef Link).

[22] Jiang, B., Luo, R., Mao, J., Xiao, T., & Jiang, Y, "Acquisition of localization confidence for accurate object detection," in *Proc. of the European Conference on Computer Vision (ECCV 2018)*, Munich, Germany, vol. 11218, pp. 816-832, 2018. Article (CrossRef Link).

[23] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12993-13000, 2020. Article (CrossRef Link).

Yao Fan is working as an Associate Professor in School of Information Engineering, Xizang Minzu University, Xian Yang, China. Her research interests include the digital protection of Tibet culture heritage and image processing.

Yubo Li is currently a master student in the School of Information Engineering, Xizang Minzu University, Xian Yang, China. His research interests include image processing and computer vision.

Yingnan Shi is currently a master student in the School of Information Engineering, Xizang Minzu University, Xian Yang, China. His research interests include deep learning and image inpainting.

Shuaishuai wang is currently a master student in the School of Information Engineering, Xizang Minzu University, Xian Yang, China. His research interests include image processing and deep learning