Real-time object detection of optical image with a lightweight model

HUANG LVE[1], LIN CHUXIN[2], XIAO WENYAN[3, *]

[1] School of Electrical Engineering and Automation, Jiangxi University of Science and Technology,
 Ganzhou 341000, Jiangxi, China

[2] School of Science, Jiangxi University of Science and Technology,
 Ganzhou 341000, Jiangxi, China

[3] School of General Education, Jiangxi University of Science and Technology,
 Nanchang 330022, Jiangxi, China

*Corresponding author: wy.xiao@jxust.edu.cn

To process the massive optical image data collected in machine vision systems and address the limitations of current learning detection models for real-time processing, this paper proposes a lightweight and real-time detection model based on YOLOX-Nano. While YOLOX-Nano is a lightweight object detection model, its detection accuracy is relatively low. Thus, this paper focuses on ensuring a lightweight model while maintaining high accuracy. The improved model incorporates an attention mechanism based on spatial and channel features to enhance the feature extraction capability of the YOLOX-Nano model. Additionally, a dual decoupled feature fusion approach is introduced to further improve the weighted fusion of feature maps extracted at different levels. This approach addresses the issue of smaller objects being overlooked in multi-object detection and enhances detection accuracy. Compared with the YOLOX-Nano baseline model, the proposed model achieves a detection speed of 59.52 FPS (frames per second) while increasing the AP50:95 metric. It meets the requirements for real-time detection, which is suitable for deployment on embedded systems, enabling the requirements of miniaturized optical processing tasks.

Keywords: lightweight optical detection system, YOLOX-Nano, decoupled attention mechanism (DAM), dual decoupled feature fusion (DDFF).

1. Introduction

Real-time optical image processing systems are widely used in fields such as aerospace, medicine, education, agriculture, industry, security, and other fields. It consists of an optical collector, processor, and image processing algorithm, among which the image processing part is the key to real-time systems. In the early stage, we constructed a real-time image acquisition preprocessing algorithm [1]. This paper continues to

study how to process optical images in real time, that is, how to build a real-time object detection algorithm with general hardware resources.

Addressing the substantial volume of collected images to achieve high stability and high-frequency detection in computer vision is crucial for enhancing the processing speed of object detection. LIU *et al*. [2] introduced the EA edge feature for object detection, utilizing EA-HOG to generate two symmetric images, which improved detection precision. However, this falls short of the real-time requirement of over 30 frames per second. WEI *et al*. [3] combined the Harr feature with HOG for detection, achieving a detection speed of 137 ms per frame. ZHAO [4] proposed a method combining a sliding window with HOG and then detecting with SoftMax, attaining a detection speed of 27 FPS. Feature extraction is a critical step in object detection, yet the computation speed of HOG features is slow. The aforementioned traditional methods depend on image localization for feature extraction, and the absence of weakly supervised methods significantly increases computational complexity. Additionally, the detection speed remains relatively low, limiting real-time detection of massive images.

Deep learning-based methods can swiftly explore the optimal solution through self-learning, enhancing real-time image detection. In the field of computer vision systems, processing massive image data increasingly relies on deep learning-based object detection algorithms [5]. WEI *et al*. [6] used an improved YOLO V3 algorithm to detect objects, implementing an unsupervised neural network to extract and classify object features. This approach achieves a detection speed of 58.4 FPS, significantly improving the aforementioned traditional methods. GAO *et al*. [7] proposed the CSPDarknet53 residual block embedded in the YOLO V4. This network model, with a size of 218.2 MB, offers robust detection capabilities. However, the large number of model parameters poses challenges for deploying in embedded machine vision systems.

GE *et al*. [8] proposed the YOLOX algorithm, a high-performance anchor-free detector that significantly reduces the number of model parameters. To adapt to miniaturized systems and minimize model parameters, GE *et al*. designed YOLOX-Tiny, and YOLOX-Nano. YOLOX-Tiny has a 10.1% improvement in AP while reducing the parameters compared to YOLO v4-Tiny. JI *et al*. [9] used YOLOX-Tiny as a benchmark, combining the convolutional block attention module with an adaptive spatial feature fusion strategy. They developed an apple object detection method based on Shufflenetv2-YOLOX, achieving 65 frames per second while maintaining high average detection accuracy. This method is also applicable to detecting other objects. While the attention mechanism aids in detecting small and medium objects, it may not enhance each effective feature layer of the feature pyramid, leading to over-coupling issues. Additionally, the YOLOX-Tiny is not the smallest model in terms of parameter volume. YOLOX-Nano, though suitable for real-time object detection, suffers from lower accuracy and precision, with higher object leakage and misdetection rates. In summary, the research goal of this paper is to improve the accuracy of lightweight models and design a real-time detection model with a small parameter volume.

Based on the YOLOX, LIU *et al*. [10] proposed an Adaptive Feature Pool, which connects the feature net and all the feature levels. For one-stage object detection based on

a feature pyramid, the inconsistency between features of different scales is one of the main limiting factors. LIU *et al*. [11] proposed adaptively spatial feature fusion (ASFF), an adaptive fusion strategy that realizes the spatial fusion of feature maps of different scales. HU *et al*. [12] applied ASFF to fuse feature maps of different sizes in the feature pyramid and used the coordinated attention mechanism to further improve the feature extraction, obtaining a detection speed of 54.35 FPS. The attention mechanism allows the model to selectively focus on specific parts of the information, thus alleviating information overload.

Lightweight detection models often struggle to achieve high accuracy due to their small number of parameters. However, their architecture is similar to the complex networks, allowing for compensation in feature extraction through feature enhancement and feature fusion. Structural expansion, though, leads to larger model sizes and increased training time.

To address the challenge of low detection accuracy for both larger and smaller objects, this paper utilizes the YOLOX-Nano model combined with the lightweight attention mechanism module since different sizes of feature maps are suitable for detecting different sizes of objects. This approach realizes the fusion of feature maps by splicing and weighting, and solves the problem of inconsistent feature scales in the feature pyramid, improving the model's detection accuracy. Thus, it enhances the detection accuracy under the same amount of the advanced model volume.

2. Lightweight model construction

The improved model proposed in the paper is shown in Fig. 1, which contains three parts: CSPDarknet, feature pyramid networks (FPN) and YOLO head. CSPDarknet and FPN form a feature extraction network, while the YOLO head is a detector specific to the YOLOX algorithm. CSPDarknet and FPN constitute the backbone of YOLOX, *i.e.*, the feature extraction network of YOLOX. In this paper, for the effective feature layers extracted in the backbone, a decoupled attention mechanism (DAM) based on spatial and channel features is proposed to realize the decoupling to achieve targeted feature enhancement.

FPN is an enhanced feature extraction network for YOLOX, where three effective feature layers obtained from the backbone part are fused. To solve the inconsistency problem within the feature pyramid, a dual decoupled feature fusion (DDFF) is designed to optimize the matching of the feature map.

2.1. Decoupled attention mechanism model

Aiming at the characteristic of the backbone network to extract feature maps with multiple resolutions and channel numbers, this paper designs a decoupled attention mechanism based on spatial and channel features, as shown in Fig. 1. The decoupled attention mechanism can purposefully strengthen the feature extraction ability of the pyramid at different layers. Accordingly, based on the classification method, we de-
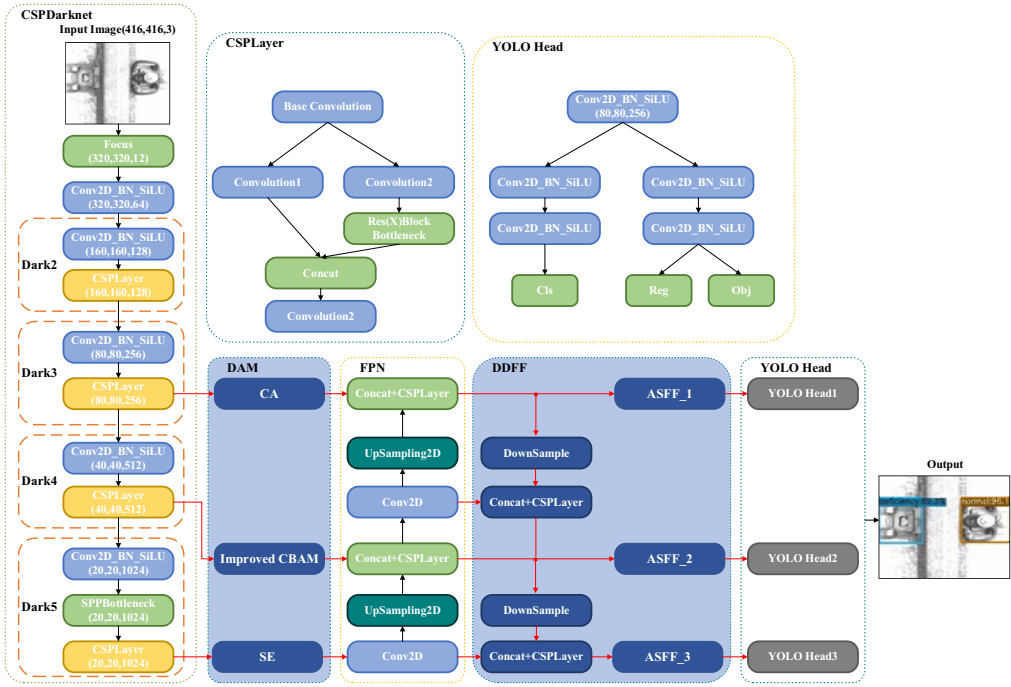
Fig. 1. Improved YOLOX-Nano network architecture.

signed an attention mechanism suitable for spatial and channel features respectively, achieving strong decoupling of spatial and channel features.

2.1.1. The SE module

The squeeze and excitation (SE) [13] attention module network architecture is shown in Fig. 2, where r is the scaling rate.
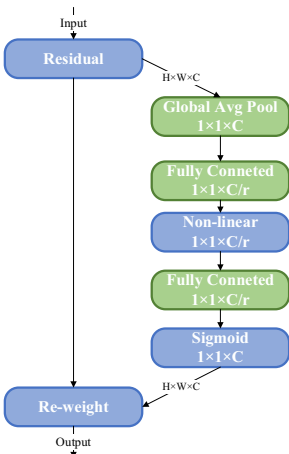

Fig. 2. Squeeze and excitation module.

First, after the basic convolution calculation, the feature maps of size u_c are obtained, and the c feature map $H \times W$ are obtained, and the feature map u_c is calculated as follows,

$$u_c = v_c * X = \sum_{s=1}^{c'} v_c^s * X^s \tag{1}$$

where v_c denotes the convolutional kernel, X^s denotes the input, and $X \in R^{W \times H \times C}$. The squeeze operation is then performed, *i.e.*, global average pooling is used to transform the input $W \times H \times C$ into the output $1 \times 1 \times C$.

$$Z_c = \frac{1}{W * H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i, j) \tag{2}$$

$$s = F_{ex}(z, W) = \sigma(g(z, w)) = \sigma(W_2 \delta(W_1 * z)) \tag{3}$$

The final excitation operation is as Eq. (3), where $W_1 * z$ is the fully connected operation and δ is the ReLU activation function, which ultimately yields the weights s of the feature maps of each channel.

2.1.2. Improved CBAM

The convolutional block attention module (CBAM) [14] can sequentially generate attention feature map information in both channel and spatial dimensions, and then the information of both feature maps is multiplied with the previous original input feature map for adaptive feature correction to produce the final feature map.

For the feature maps generated by the backbone network extraction,

$$F \in R^{C \times H \times W} \tag{4}$$

CBAM produces one-dimensional maps of channel attention features, respectively,

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \tag{5}$$

$$M_c \in R^{C \times 1 \times 1} \tag{6}$$

where σ is the sigmoid activation function. Further, the spatial attention feature map is computed in two dimensions,

$$M_s(F) = \sigma(f^{7 \times 7}(\text{AvgPool}(F); \text{MaxPool}(F))) \tag{7}$$

$$M_s \in R^{1 \times H \times W} \tag{8}$$

and $f^{7 \times 7}$ is a 7×7 convolution operation and element-wise multiplication of (2) and (4) in turn,

$$F' = M_c(F) \otimes F \qquad (9)$$

$$F'' = M_s(F') \otimes F' \qquad (10)$$

In the above equation $\otimes$ denotes element-level multiplication, with a broadcast mechanism for dimensional transformation and matching in between. CBAM is realized based on the channel attention mechanism and spatial attention, and to strengthen the role of the channel attention mechanism, SE is used to replace the traditional channel attention mechanism in CBAM. Therefore, Eq. (10) is modified as follows, SE_s is F' the feature map extracted by SE,

$$F'' = SE_s(F') \otimes F' \qquad (11)$$

2.1.3. CA module

The coordinate attention (CA) mechanism [15] (as shown in Fig. 3) encodes the exact location from the height and width of the image and obtains the feature maps in both width and height directions, respectively,

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W}^{1} x_c(h, i) \qquad (12)$$

$$Z_c^W(W) = \frac{1}{H} \sum_{0 \le j \le H}^{1} x_c(j, W) \qquad (13)$$

The feature maps in both full-width and height directions are obtained to be stitched together by the concat operation, which is sequentially computed by the shared con-
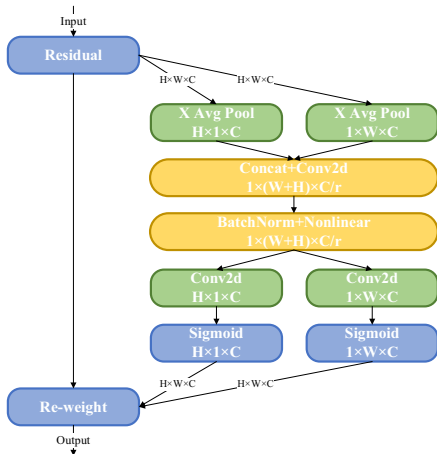

Fig. 3. Coordinate attention module.

volution kernel as a 1×1 convolution, the batch normalization process, and the sigmoid activation function,

$$f = \delta(F_1[z^h, z^w]) \tag{14}$$

The feature maps F_h and F_w are obtained by convolving the height and width of f with a convolution kernel of 1×1, respectively, and the attention weights of the feature maps on the height and width, g^h and g^W, are calculated by sigmoid.

$$g^h = \sigma(F_h(f^h)) \tag{15}$$

$$g^w = \sigma(F_w(f^W)) \tag{16}$$

Finally, the final feature map with attentional weights in the width and the height directions will be obtained by multiplicative weighting calculation on the original feature map with the formula shown below,

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^W(j) \tag{17}$$

2.2. Dual decoupled feature fusion

As included in Fig. 1, the feature maps are up-adopted and down-sampled, then spliced to form a simple combination, which fails to fully reflect the differences between the feature maps. In CNN, the shallow layer is weak in semantic information but rich in location information. Conversely, deeper layers possess stronger semantic information but weaker location information. In the original FPN, when a feature map matches an object, the information on the feature maps of other layers is ignored. To address this, the first weight of feature fusion, after twice downsampling to achieve consistent feature map resolution size, combined with CSPLayer+Concat operation to achieve splicing.

There is a network structure similar to a fully connected layer between the ASFF and the output. This structure allows the model to adaptively train and get the weight sizes of the three feature layers, enhancing object detection. However, this process increases the number of model parameters.

$$y_{ij}^l = \varphi_{ij}^l * x_{ij}^{1 \to l} + \beta_{ij}^l * x_{ij}^{2 \to l} + \gamma_{ij}^l * x_{ij}^{3 \to l} \tag{18}$$

Take the third fused feature ASFF-3 as an example. The vector at the spatial location (i, j) after fusion is the weighted fusion of the vectors at the three feature maps (i, j) before fusion, and $i, j \in (0, 3)$. x_1, x_2, x_3 are the feature maps from the three different layers. By multiplying the weight parameter $\varphi_1, \beta_2, \gamma_3$ by the features from the different layers and adding them up, we can obtain a new fused feature ASFF-3. Due to the summing method, the output features from different layers must have the same size and the same number of channels. Therefore, it is necessary to upsample or downsample

the different features and adjust the number of channels to match the output of the first feature map.

$$\varphi_{ij}^{l} = \frac{\exp(\lambda_{\alpha_{ij}}^{l})}{\exp(\lambda_{\alpha_{ij}}^{l}) + \exp(\lambda_{\beta_{ij}}^{l}) + \exp(\lambda_{\gamma_{ij}}^{l})} \tag{19}$$

The coefficients λ in Eq. (19) are the spatial importance weights of the feature map, which are adaptively learned by the network and they are shared among all channels. The coefficients λ are assigned as weights to different feature layers, and these weights are self-adaptively trained by the BP algorithm using 1×1 convolution to distinguish the importance of different feature layers. The parameters φ, β, γ are passed through softmax after concat operation, ensuring that they all fall within the range $[0, 1]$ and sum to 1.

3. Experimental results and comparative analysis

3.1. Data preprocessing

Due to the large kinds of objects involved in object detection, we chose the railway track fasteners as the object in this experiment. Defects or missing fasteners may lead to safety accidents, so it is necessary to process the collected fasteners images in real -time [16]. HU *et al*. classified all the defects fasteners into three categories: normal, crack, and displacement [12]. However, they did not consider the defects cases of missing fasteners and deformation of strips, which may limit the object detection model's generalization ability. In this paper, we further subdivide the defects categories and increase the training samples to seek better generalization ability. The models mentioned in the previous section [6, 7, 12] were trained on less than 2,000 samples, which poses a risk of overfitting when the number of samples is too small. To address this problem, we collect more samples of fasteners to ensure sufficient training and avoid overfitting.

Based on the collected samples and data enhancement, a total of 7,939 samples, including nearly 10,000 detection objects, were finally obtained. These were classified into six categories: normal, crack, deficiency, displacement, cover, and deformation. The samples were divided into the training set, test set, and validation set according to the approximate average distribution ratio. The number of samples included in each category is shown in Table 1. Investing in a large number of training samples can make the model get repeated training, ensuring high accuracy and strong generalization ability while avoiding overfitting.

T a b l e 1. Sample size of different types of image data.

Dataset	Normal	Crack	Deficiency	Displacement	Cover	Deformation
Train	258	860	1664	1546	283	152
Test	94	305	521	516	110	22
Val	100	309	515	536	86	22

3.2. Comparison of training results and performance

The max epoch is set to 500 and batch size is set to 32. In this paper, we use control variables to compare the differences before and after the model improvement for the dataset. YOLOX-Nano is used as the control group, while our improved YOLOX-Nano is used as the experimental group.

The validation set is then used to calculate the mean average precision (mAP) for each category of object detection. The mAP values reflect the comprehensive performance of the detection model in recognizing detections across all categories.

As shown in Table 2, the number of parameters of the improved model in this paper is 2.31M. This represents a slight increase compared to the model with the ASFF, but it is still significantly smaller than the YOLOX-Tiny [8] of 5.06M, which is consistent with the lightweight design. The actual increase in floating-point computation is about 1.99G, which remains within the acceptable range.

AP50:95 is the result of the YOLO algorithm under the IoU = 0.5 condition, where AP is calculated at intervals of 0.05, and the overall average is finally calculated. The training results of YOLOv5n, YOLOX-Nano, YOLOX-Tiny, and faster R-CNN are included for comparison in this paper.

The model detection performance before and after improvement is shown in Table 3. The faster R-CNN detection rate using MobilenetV2 with 3.4M [19] parameters as the backbone is significantly higher than that of the Faster R-CNN with Resnet50+FPN selected as the backbone. The number of parameters of Resnet50 is about 25.64M, and the number of parameters of FPN is significantly higher than that of MobilenetV2.

T a b l e 2. Our improved YOLOX-Nano parameters and computation volume.

Model	Params(M)	Flops(G)
YOLOX-Nano	0.9	1.08
CBAM-YOLOX-Nano	0.92	1.08
ASFF- YOLOX-Nano	2.27	1.99
ASFF&CBAM-YOLOX-Nano	2.29	1.99
DDFF&DAM-YOLOX-Nano	2.31	1.99
Ours	**2.31**	**1.99**

T a b l e 3. Detection performance of different models.

Model	Backbone	AP50:95 [%]	Frame per second
Faster R-CNN [17]	FPN+Resnet50 [18]	45.6	20.73
Faster R-CNN	MobilenetV2 [19]	34.4	53.58
YOLOv5n	CSPDarknet	79.1	52.06
YOLOX-Tiny	CSPDarknet	81.3	58.48
YOLOX-Nano	CSPDarknet	81.12	59.88
ASFF&CBAM-YOLOX-Nano	CSPDarknet	81.46	57.14
DDFF&DAM-YOLOX-Nano	CSPDarknet	82.08	58.13
Ours	**CSPDarknet**	**82.72 (+2.6)**	**59.52**

The model with more parameters has a slower detection rate, which aligns with the characteristics mentioned in the previous section. However, this approach has lower detection accuracy, a common deficiency of lightweight neural network models. Despite this, the detection speeds of both Faster R-CNNs with these two different backbones are lower than YOLOX, and their detection accuracies are also lower.

Compared with the original YOLOX-Nano, our method has a 1.6% improvement in AP50, with no significant increase in average inference time. Moreover, the number of parameters is much smaller than that of YOLOX-Tiny while improving the detection accuracy, effectively balancing the number of parameters and accuracy. Comparing the original algorithm with ASFF&CBAM-YOLOX-Nano and the method in this paper, it can be seen that YOLOX combined with ASFF can reduce the model instability caused by the differences in the scale of the feature maps, leading to higher detection accuracy. The feature fusion mechanism significantly improves real-time detection speed, while the attention mechanism enhances the feature extraction ability for small objects, thus improving overall detection accuracy. Therefore, the combination of ASFF & CBAM-YOLOX-Nano with CBAM and the method in this paper performs better in the performance index of AP50:95. The DAM used in our method adopts an objected attention mechanism for the characteristics of each layer of the feature pyramid, which realizes the decoupling of the feature reinforcement method and shows superior detection performance. This method utilizes the idea of partitioning, different from the multi-layer superposition of the attention mechanism on the feature map, effectively improving detection performance without generating more parameters and computation.

In terms of time expenses, our method achieves a detection speed of 59.52 FPS while also increasing the detection accuracy, meeting the requirements for real-time efficient detection. This demonstrates that our method improves the detection performance of the model while maintaining reasonable control over time and space overhead, resulting in an accurate and effective real-time detection model.
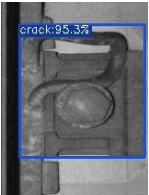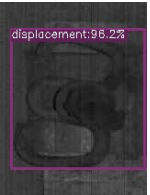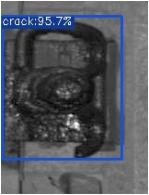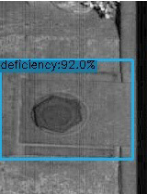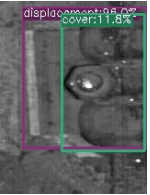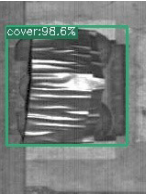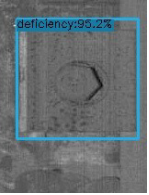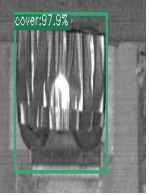
3.3. Training parameter settings

The maximum value of batch during training is set to 32 to avoid memory overflow. The attenuation coefficient is 0.0005, and the image size is standardized to (416, 416) for training. The activation function chosen is silu. Following the training method proposed in the original YOLOX article, we removed the mixup and weakened the mosaic effect for smaller models, which has been experimentally shown to acquire better training results [8]. Therefore, the optimization parameter mosaic-scale is adjusted from (0.1, 2) to (0.5, 1.5) to weaken the mosaic effect in training.

3.4. Optical image detection in real time

Table 4 shows the effect of our model on detecting defects in six types of railway fasteners. Compared with the performance of various detection methods mentioned above, the confidence level for small object classification is slightly lower. However,

T a b l e 4. Detection performance in different fastener object defects.

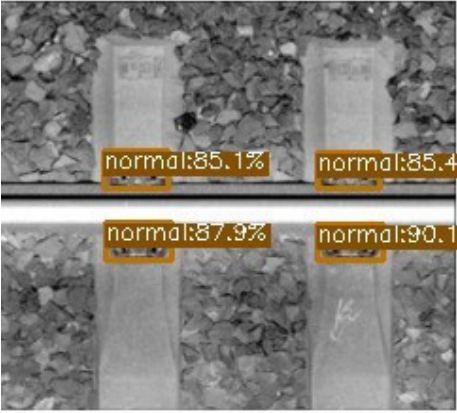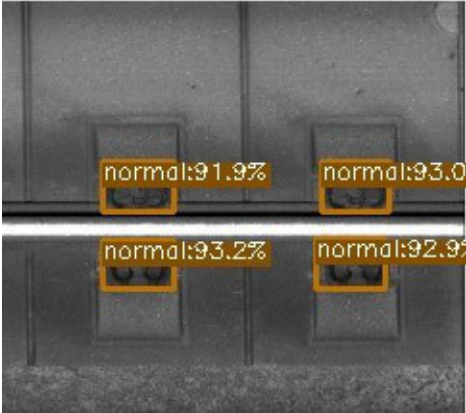|	Normal	Crack	Deficiency	Displacement	Cover	Deformation
1	(1-1)	(2-1)	(3-1)	(4-1)	(5-1)	(6-1)
2	(1-2)	(2-2)	(3-2)	(4-2)	(5-2)	(6-2)
3	(1-3)	(2-3)	(3-3)	(4-3)	(5-3)	(6-3)

Fig. 4. Real-time detection of the small optical object on ballasted track and no-ballasted track.

the confidence level of our algorithm's detection results is above 85%, as shown in Fig. 4, with no classification judgment errors. Therefore, our model also demonstrates high accuracy in detecting small targets.

4. Conclusions

The proposed method reduces the existing model's size and makes it applicable in embedded vision systems. It enhances the model's feature extraction capabilities based on spatial and channel features with the decoupled attention mechanism and addresses the issue of inconsistent object feature scale with the dual decoupled feature. Therefore, the proposed model addresses the problem of insufficient accuracy of the lightweight model in detecting small objects, achieving a balance between model volume and detection accuracy with a comparable detection speed. Further work could consider deblurring images to improve the accuracy of object detection [20, 21]. This improved model has been applied in the detection of railway track defects, and also it can be extended to other real-time object detection.

References

[1] HUANG L.E., WU L.S., XIAO W.Y., PENG Q.J., *Deblurring approach for motion camera combining FFT with α-confidence goal optimization*, Optica Applicata **50**(2), 2020: 185-198. https://doi.org/10.37190/oa200202

[2] LIU J.J., YUAN J.Y., JIA Y.F., *A new method for railway fastener detection using the symmetrical image and its EA-HOG feature*, International Journal of Pattern Recognition and Artificial Intelligence **34**(2), 2020: 2055006. https://doi.org/10.1142/S021800142055006X

[3] WEI Y., TIAN Q., GUO J.H., HUANG W., CAO J., *Multi-vehicle detection algorithm through combining Harr and HOG features*, Mathematics and Computers in Simulation **155**, 2019:130-145. https://doi.org/10.1016/j.matcom.2017.12.011

[4] ZHAO Y.G., ZHENG F., SONG Z., *Hand detection using cascade of softmax classifiers*, Advances in Multimedia, Vol. 2018, 2018: 9204854. https://doi.org/10.1155/2018/9204854

[5] HUANG D.Q., FU Y.Z., QIN N., GAO S.B., *Fault diagnosis of high-speed train bogie based on LSTM neural network*, Science China Information Sciences **64**(1), 2021: 119203. https://doi.org/10.1007/s11432-018-9543-8

[6] WEI R.Y., LI S.T., WU S.R., *Defect detection of track fastener based on improved YOLO V3 algorithm*, Railway Standard Design **64**(12), 2020: 30-36.

[7] GAO J.L., BAI T.B., YAO D.C., *et al*., *Detection of track fastener based on improved YOLOv4 algorithm*, Science Technology and Engineering **22**(7), 2022: 2872-2877.

[8] GE Z., LIU S.T., WANG F., LI Z.M., SUN J., *YOLOX: Exceeding YOLO series in 2021*, 2021, arXiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430

[9] YAN B., FAN P., LEI X.Y., LIU Z.J., YANG F.Z., *A real-time apple targets detection method for picking robot based on improved YOLOv5*, Remote Sensing **13**(9), 2021: 1619. https://doi.org/10.3390/rs13091619

[10] LIU S., QI L., QIN H.F., *Path aggregation network for instance segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, USA: 8759-8768.

[11] LIU S., HUANG D., WANG Y., *Learning spatial fusion for single-shot object detection*, 2019, arXiv:1911.09516. https://doi.org/10.48550/arXiv.1911.09516

[12] Hu J., Qiao P., Lv H., Ouyang A., He Y., Liu Y., *High speed railway fastener defect detection by using improved YoLoX-Nano model*, Sensors (Basel) **22**(12), 2022: 8399-8415. https://doi.org/10.3390/s22218399

[13] Hu J., Shen L., Sun G., *Squeeze-and-excitation networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, USA: 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[14] Woo S., Park J., Lee J.Y., Kweon I.S., *CBAM: Convolutional block attention module*, [In] Ferrari V., Hebert M., Sminchisescu C., Weiss Y. [Eds.] *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, Vol. 11211, Springer, Cham. https://doi.org/10.1007/978-3-030-01234-2_1

[15] Hou Q.B., Zhou D.Q., Feng J.S., *Coordinate attention for efficient mobile network design*, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021: 13713-13722. https://doi.org/10.1109/CVPR46437.2021.01350

[16] Zhang X., Liu Z., *Fast color image encryption algorithm based on FCSM and pre-storage Arnold transform*, Multimedia Tools and Applications **83**(2), 2024: 3985-4016. https://doi.org/10.1007/s11042-023-15577-6

[17] Ren S.Q., He K.M., Girshick R., Sun J., *Faster R-CNN: Towards real-time object detection with region proposal networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 2017: 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[18] He K.M., Zhang X.Y., Ren S.Q., Sun J., *Deep residual learning for image recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016: 770-778.

[19] Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C., *MobileNetV2: Inverted residuals and linear bottlenecks*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018: 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[20] Zhou N., Deng J., Pang M., *Recovering a clean background: A parallel deep network architecture for single-image deraining*, Pattern Recognition Letters **178**, 2024: 153-159. https://doi.org/10.1016/j.patrec.2024.01.006

[21] Zhang H.Y., Wang Y., Dayoub F., Sünderhauf N., *VarifocalNet: An IoU-aware dense object detector*, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021: 8510-8519. https://doi.org/10.1109/CVPR46437.2021.00841