A Lightweight Convolutional Neural Network For Real-time Detection Of Aircraft Engine Blade Damage

Wenzhe Wang, Hua Su, Xinliang Liu, Jawad Munir, and Jingqiu Wang*

National Key Laboratory of Helicopter Aeromechanics, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China

Corresponding author. E-mail: meejqwang@nuaa.edu.cn

Received: Jul. 24, 2024; Accepted: Sep.12, 2024

To address the large number of parameters and the computational complexity of deep learning models in the field of borescope detection, we propose a lightweight blade damage detection model LSSD using a knowledge distillation algorithm. First, the inverse residual structure is used to lightweight the backbone network of the classic SSD model. Then, the K-means clustering algorithm is used to optimize the scale and number of anchor boxes to reduce the parameters and computational complexity of the proposed model. Second, to ensure that the lightweight model has a certain level of detection accuracy, a feature fusion module CA-FPN combined with coordinate attention and a small damage detection enhancement module W-Inception are embedded. Finally, the knowledge distillation algorithm is used to further improve the detection accuracy of the model. The number of parameters of the LSSD model is 4.99M, the MACs is 3.541G, and the detection speed reaches 32FPS. Compared with the SSD model, the LSSD model reduces the number of parameters by 79.3% and the computational complexity by 88.42%, resulting in a 2-fold increase in the detection speed.

Keywords: Aero-engine blade, Damage detection, Borescope detection, Lightweight convolution neural network, Knowledge distillation

http://dx.doi.org/10.6180/jase.202508_28(8).0013

1. Introduction

As the power heart of an aircraft, the aeroengine is the basic guarantee for safe flight. Blades are key components of aeroengines, and they are prone to damage, such as ablation, notches, and cracks, under harsh working conditions such as high temperature, high pressure, and variable loads [1–4], posing a serious threat to aircrafts. Omitting defects during the inspection process can cause serious damage to the engine and aircraft and may lead to serious accidents [5]. Borescope detection, the most widely used blade damage detection technology at present [6, 7], can realize in situ blade damage detection and avoid the complicated work of disassembling an engine. However, borescope mirrors mainly rely on manual operation and have uncertain factors due to personnel differences. In the face of various types and sizes of damage and poor lighting conditions, they are easily missed or misreported. To improve the automation and intelligence of borescope detection, many researchers have introduced image processing[8, 9] and deep learning technology. Wong et al. [10] used the Mask RCNN model to segment and track damage to rotor blades. Cai et al. [11] proposed a lightweight YOLOv4 model to solve the problems of a large number of parameters and slow detection speed. At the cost of losing 3.55% of the mAP, the model parameters were reduced to one-third of the original values, resulting in an average detection speed of 37.3 fps. He[12] proposed a SW-YOLO model for borescope video detection of aeroengine blades. The model added a spatial channel attention module to the backbone network of YOLOv5 and optimized the feature pyramid used for feature fusion in the neck network, improving

the detection accuracy of the model. Li[13] used DDSC-YOLOv5 for aircraft engine borescope detection video and introduced variable convolution on the basis of YOLOv5 to overcome the disadvantage of poor geometric transformation of neural networks. They also used variable convolution to improve the efficiency of feature extraction while reducing computational complexity and improving accuracy while minimizing computational complexity. Shang[14] constructed a network model with classification, localization, and segmentation functions based on Mask R-CNN for texture information of damage and proposed practical evaluation indicators for blade damage detection. Wei[15] designed a remote borescope detection system that works in collaboration with a borescope, Raspberry Pi, and server, which has significant advantages in terms of damage detection accuracy, real-time performance, and maintenance costs. The video collected by the borescope is encoded and processed frame by frame by Raspberry Pi and then transmitted to a remote server. The server completes damage detection and sends the results back to Raspberry Pi for display through the local area network.

Due to resource constraints in industrial production, lightweight networks often perform better in engineering practice. Howard [16] proposed a lightweight neural network called MobileNets for embedded devices such as mobile phones by effectively reducing network parameters through convolutional kernel decomposition. The shuffleNet[17] model proposed by Zhang et al. uses point by point convolution and channel shuffling, greatly reducing the computational complexity of the network, and performs well on ImageNet and MS COCO. Liu[18] et al. proposed the deep learning model YOLOv3-Mobilenet-PK using backbone network replacement, filtering pruning, and knowledge distillation methods, and successfully deployed it on edge devices to detect surface damage of the WT blades.

The above deep learning models used for borescope detection have high accuracy, but they have the characteristics of a large number of parameters, high computational complexity, and high hardware configuration requirements, making it difficult to directly deploy them in mobile devices with limited hardware resources. Therefore, we propose a lightweight LSSD model based on knowledge distillation for blade damage detection, to achieve integrated intelligent damage detection deployed on edge devices.

2. Methods

SSD is an excellent one-stage object detection algorithm proposed by Liu et al. [19]. The SSD object detection algorithm combines the advantages of Faster-RCNN[20] and YOLO[21] algorithms, effectively balancing detection accuracy and speed, and is widely used in the fields of surface defect detection and damage detection. Therefore, we improved and optimized the SSD model, and constructed a lightweight convolutional neural network model LSSD for aircraft engine blade damage detection. The structure of LSSD model is shown in Fig. 1. The main improvements of LSSD Model include: (1) the backbone network of LSSD was reconstructed using the inverse residual structure, the number and size of anchor boxes were optimized using K-means clustering method, resulting in a significant reduction of the model parameter and computational complexity; (2) by integrating the improved feature fusion structure CA-FPN and the W-Inception module designed for detecting small-sized blade damage, ensure LSSD model has gratifying detection accuracy, (3) using knowledge distillation strategy to improve the detection capability of the model without increasing model parameters.

2.1. Backbone network lightweight

By applying deep learning technology to achieve intelligent borescope detection on mobile devices with limited hardware resources, the convolutional neural network models should have the characteristics of fewer parameters and lower computational complexity. MobileNetv2 not only reduces model parameters and computational complexity, but also gives good consideration to accuracy. Its excellent structural design can be well transferred to our model. Therefore, we replaced VGG16 in the SSD model backbone network with MobileNetv2[22] and extended a new feature layer with an inverted residual structure.

We utilized the core idea of MobileNetv2, which is depth separable convolution and inverse residual structure, to construct the backbone of LSSD. Depth separable convolution decomposes the convolution into channel wise convolution and point wise convolution, significantly reduce the model parameter and computational complexity. In LSSD, we perform point by point convolution to increase the dimensionality of the input feature map, then use channel by channel convolution with a kernel size of 3×3 to extract features from the feature map. Finally, perform point by point convolution to reduce the dimensionality of the feature map. The entire process is characterized by an inverted residual structure of "expansion-convolution-compression", ensuring that lightweight network has high detection accuracy.

2.2. Optimization of initial values of anchor boxes

The anchor boxes of SSD model is designed for the detection of targets such as people, cars, and other common

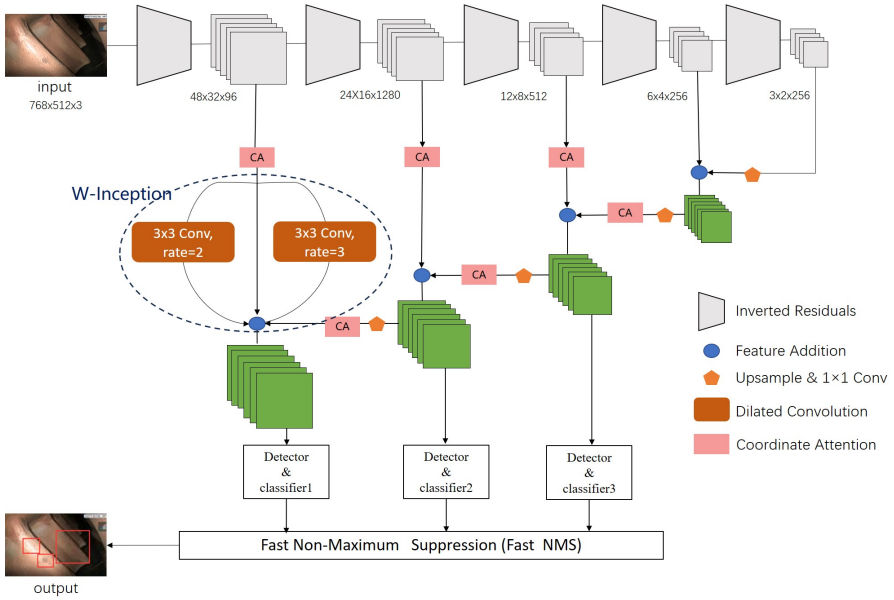


Fig. 1. Structure of the LSSD model

objects in daily life. However, LSSD is designed for the detection of the blade damages such as ablation, burn-through, notch, and crack. The objects in the two datasets have significant differences, and using the original anchor boxes will not only introduce a large number of redundant parameters and computational load, but also slow down the convergence speed of the network. To avoid the above defects, we use the K-means clustering algorithm to cluster the ground-truth bounding boxes and reset the scale of the anchor boxes. The specific workflow is as follows:

1. Randomly select several ground-truth bounding boxes (i.e. $k = 5$), and the width of a bounding box is w_{ki}, the height is h_{ki}, and they are used as cluster centers.

2. Calculate the distance from all ground-truth bounding boxes in the aircraft engine blade damage dataset to k cluster centers, and mark each ground-truth bounding box as the same class as the nearest cluster center.

3. Calculate the average width and height of the ground-truth bounding box marked as the same class in (2), and regenerate k cluster centers.

4. Repeat steps (2) and (3) until the coordinates of k cluster centers no longer change.

Use Euclidean distance to measure the distance between each ground-truth bounding box and k cluster centers. The calculation formula for Euclidean distance is as follows:

$$D\left(w,h,w_{ki},h_{ki}\right) = \sqrt{\left(w-w_{ki}\right)^2 + \left(h-h_{ki}\right)^2} \quad (1)$$

where w is the width of any box in the dataset, h is the height of any box in the dataset. The calculation formula for regenerating k cluster centers is:

$$m'_j = \frac{1}{N_j}\sum m \quad (2)$$

where m represents any coordinate with m_i as the cluster center, N_j represents the number of anchor boxes in the same class as m_i (including m_j), and m'_j represents the j-th new cluster center.

The process of resetting anchor boxes using K-means clustering algorithm is shown in the Fig. 2.

2.3. Feature fusion structure

The shallow feature map of the SSD model only contains edge information, which has insufficient representation ability. Moreover, when the backbone network is lightweight, the network's ability to capture spatial information deteriorates, resulting in a significant decrease in detection accuracy. To address the above issues, we introduce the improved feature fusion structure CA-FPN into the LSSD model. The FPN structure[23]transfers the semantic information of deep networks to shallow networks, giving shallow feature maps rich feature information and enhancing the model's detection ability for small and medium-sized targets. Coordinate attention can capture the cross-channel and positional information of feature layers, allowing the model to more accurately identify and locate blade damage. Coordinate attention[24] is a novel and efficient attention mechanism proposed by Hou in

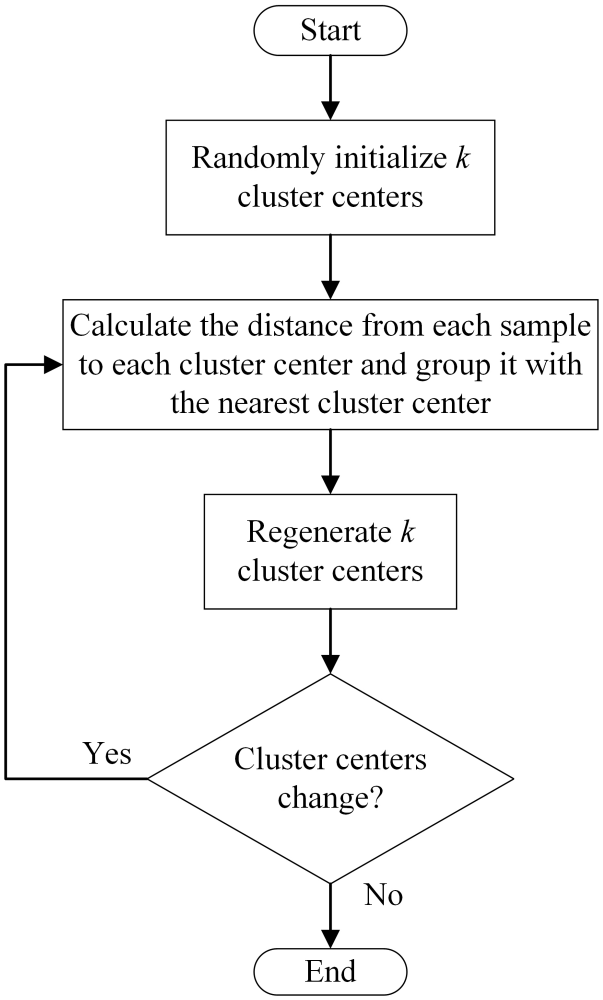


Fig. 2. Reset anchor box process using K-means clustering algorithm

2021. It simultaneously performs global pooling on the input feature map along the H and W directions, generating two one-dimensional feature vectors with directional awareness. The feature vectors are convolved and encoded to generate an attention map that can capture long-distance dependencies. By multiplying the attention map with the input feature map, the model's attention to the location of the region of interest is strengthened. By increasing the coordinate attention mechanism, the attention of different scale feature fusion can be improved, so that the three-scale feature map output by CA-FPN can provide more useful features for the detection head.

2.4. Small damage detection enhancement module

The shallow feature maps in the SSD model have a higher resolution, thus retaining considerable texture information. However, the number of convolutions from the input image to the shallow feature map is relatively small, resulting in poor representation ability of the shallow feature maps and poor generalization ability of the model for small-scale damage detection. In order to detect small-scale damage through shallow feature maps and improve the generalization ability, we design an enhancement module, namely W-Inception, as shown in Fig. 1.

The W-Inception module includes multiscale dilated convolution kernels and employs parallel convolution, as well as residual connections, all of which are aimed at improving model performance. The specific workflow of the W-Inception module is as follows: the input feature map is parallelly convolved through two dilated convolutions with different rates; then the resulting output is fused with the original input to obtain the final output. Dilated convolution can expand the receptive field under the condition of unchanged feature map resolution, making shallow feature maps have a certain perception of global features and improving the model's detection accuracy for small damage. Residual connections can preserve important information from the input feature map and compensate for the shortcomings of dilated convolution. The fusion of feature maps adopts the operation of adding feature maps instead of concatenating feature maps, reducing the parameter and computational complexity of the proposed model and strengthening important features and suppressing background information.

2.5. Distillation of intermediate layer features

Knowledge distillation[25] is a convolutional neural network training algorithm based on the "teacher-student network concept" proposed by Hinton in 2015, which is a model compression method. The model with more parameters and a complex network structure is called the teacher network and, the model with fewer parameters and a relatively simple network structure is called the student network. The neurons, intermediate feature layers, network output values, etc. are called knowledge. The core idea of knowledge distillation is to use a pre trained deep learning model with a large number of parameters and complex structures to assist in the training of lightweight deep learning models. This means that the knowledge of the teacher network is transferred to the student network, thereby improving the generalization ability of lightweight models. Therefore, the knowledge distillation can obtain better network weights and improve the generalization ability and detection accuracy of LSSD without increasing the number of parameters and computational complexity.

We use SSD as a teacher network, which has stronger feature extraction ability and can learn more feature rep-

resentations of engine blade images. Its effective feature layer contains richer edge and semantic information[26]. Transferring the feature representations learned from the teacher network SSD as knowledge to the student network LSSD can enhance the generalization ability of the LSSD model and improve the detection accuracy of blade damage. Figure 3 shows the structure diagram of a knowledge distillation network based on intermediate layer features, where SSD is a teacher network and LSSD is a student network.

The loss function of LSSD network based on feature layer distillation consists of two parts, namely student loss and distillation loss. The calculation formula is as follows:

$$L(x, c, l, g) = L_{\text{loss}}(y, \hat{y}) + \beta L_D(s, T) \tag{3}$$

where L_{loss} represents the loss between the output of the LSSD network and the real labels, i.e. the loss function of the SSD model. L_D represents the distillation established by the teacher network SSD and the student network LSSD in the backbone network. β represents the distillation weight, which is taken as 1 here.

The calculation formula for the distillation loss is as follows:

$$MSE = \frac{\sum_{i=1}^{n} \left(y_i - y_i'\right)^2}{n} \tag{4}$$

where n is the product of the width and height of the feature layer, y_i represents the pixel value of the adaptive feature layer in the LSSD network, and $y'I$ represents the pixel value of the effective feature layer in the SSD network.

3. Results and discussion

3.1. Dataset

The proposed LSSD is trained and validated on an aero-engine blade image dataset. We take 15 frames as the interframe difference of the borescope video and use the fixed interframe difference method to extract key frames from the borescope video to create a dataset. The dataset includes blade images with four types of damage: ablation, burn-through, notch, and crack. After using image processing methods (adding noise, adjusting brightness and contrast, rotating image) for data augmentation and class balancing, a total of 3135 blade images were obtained with a resolution of 768×512, and the images were randomly divided into a training set and a test set at a ratio of 8:2.

3.2. Experimental settings

The proposed LSSD model is implemented based on the Pytorch framework using a PC platform with an i5-12600KF CPU, and the graphics card is NVIDIA GeForce RTX 3060 (12G) with 16GB of memory. During the training process, the hyper-parameters are set as follows: the training rounds are set to 300, the batches are set to 8, the initial learning rate is set to 0.002, the minimum learning rate is set to 0.00002, and the learning rate decrease method is cosine learning rate decrease. The SGD optimizer with a momentum coefficient of 0.937 is used, and the weight decay parameter of 0.0005 is applied to prevent an overfitting and obtain better results.

The commonly used evaluation indexes of target detection are: accuracy, recall, detection speed, mean average precision (mAP), parameter number, calculational cost, etc. The mAP combines two indicators, precision rate and recall rate, which can well reflect the average performance of the model in different categories, so we take the mAP, the number of model parameters, computational cost, and detection speed as the evaluation indicators of the model performance. The formulas are as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

Among them, TP is the number of positive samples that the model correctly predicts as positive, FN is the number of positive samples that the model incorrectly predicts as negative, and FP is the number of negative samples that the model incorrectly predicts as positive.

$$AP = \int_0^1 P(R)dR \tag{7}$$

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{8}$$

where K represents the number of blade damage categories, and AP represents the accuracy of the model for each type of damage.

The number of model parameters determines the size of memory or graphics space occupied by the model during operation, and is used to measure the spatial complexity of the algorithm.

Computational cost refers to the number of times a model performs numerical operations such as convolution and full concatenation, which determines the time required for model detection. It is used to measure the time complexity of the algorithm and is represented by Multiply Accumulate Operations (MACs).

The detection speed refers to the number of video frames or images that a model can detect per second. A higher FPS indicates a faster detection speed of the model. To meet the real-time detection requirements of borescope

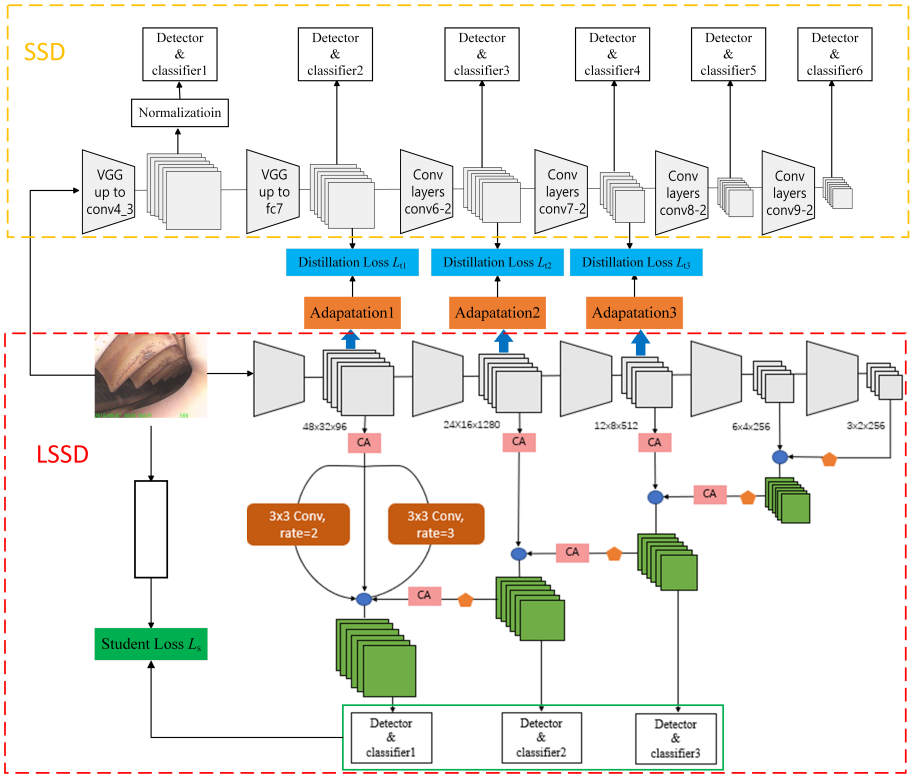


Fig. 3. Knowledge Distillation Network Structure Based on Middle Layer Features

videos, the detection speed of the model should be no less than 24 frames per second.

3.3. Ablation experiment

This section will verify the effectiveness of some improvements through ablation experiments. The improvement methods include adjusting the feature fusion structure CA-FPN, small-sized damage detection enhancement module W-Inception, and setting initial values of anchor boxes in a clustering manner.

Considering that the features of blade cracks and erosion damage are relatively small, and the original SSD compresses the image to 300x300, it may result in feature loss. Therefore, adjust the input resolution of the network to 768x512, which is the original image size. The training images used in this article are all of the original image size.

1. Feature fusion module CA-FPN
   In order to verify that the CA-FPN module can improve the feature extraction ability of the LSSD model, a comparative experiment was conducted on the test set with and without the addition of the CA-FPN module. The detection results are shown in Table 1. It can be seen that the detection accuracy of the model has been improved after adding the CA-FPN module, and the mAP has increased by 1.82%. The CA-FPN module is lightweight, so the detection speed almost unchanged.

Table 1. Comparison of damage detection results with or without CA-FPN

Model	mAP@0.5	Detection Speed
LSSD	79.30%	32 FPS
LSSD + CA-FPN	81.12%	31 FPS

2. Small size damage detection enhancement module W-Inception
   Table 2 shows the detection results of the LSSD model with and without the W-Inception module on the test set. It can be seen that after adding W-Inception to the LSSD model, the mAP increased by 1.2%, and the AP for blade ablation, burn-through, notch damage, and crack increased by 2.89%, 1.59%, 0.19%, and 0.14%, respectively. The addition of the W-Inception module expands the receptive field, and the shallow feature map has richer local feature information, effectively improving the detection accuracy of the LSSD model for small-sized damage.

3. Comparative experiments on different anchor box set-

Table 2. Comparison of damage detection results with or without W-Inception

Model	Damage Class	AP	mAP@50
LSSD + CA-FPN	Ablation	85.51%	81.12%
	Burn-through	87.06%	
	Notch	94.45%	
	Crack	63.87%	
LSSD + CA-FPN + W-Inception	Ablation	92.92%	82.32%
	Burn-through	87.33%	
	Notch	94.47%	
	Crack	69.99%	

tings

The number of anchor boxes affects the parameter and computational complexity of the model, and the size of anchor boxes affects the convergence speed of model training. Reasonable setting of the size and number of anchor boxes is crucial for improving model performance. Table 3 shows the detection results of the LSSD model on the test set with different sizes and quantities of anchor boxes. From the table, it can be seen that the performance of the model is best when setting 6 anchor boxes: the anchor boxes is set to $[8, 7]$ and $[17, 22]$ when the feature map is 38×24, set to $[47, 30]$ and $[39, 49]$ when the feature map is 38×24, set to $[69, 43]$ and $[105, 60]$ when the feature map is 38×24. In this way, the model has the mAP of the model is 82.32%, the number of parameters is 4.99 M , and the MACs is 3.54 G .

4. Comparative experiments on characteristic distillation of the intermediate layer

Fig. 3 shows the experimental results of knowledge distillation on the intermediate feature layer of the LSSD model. After distillation training, the average detection accuracy mAP of the LSSD model increased by 2.34%, the ablation damage detection accuracy increased by 1.6%, and the etching damage detection accuracy increased by 7.9%. The detection accuracy of burn-through and crack has not changed much, which is reasonable because the teacher network already has lower detection accuracy for these two types of damage. This also demonstrates the rationality and effectiveness of the distillation strategy we have adopted. Fig. 5 (a) shows the detection results of the LSSD model before distillation, Fig. 5 (b) shows the detection results after distillation, and Fig. 5 (c) shows the detection results of the SSD network. It can be seen that after knowledge distillation, the LSSD model reduces the missed detection rate of ablation and notch, and the detection accuracy is improved.

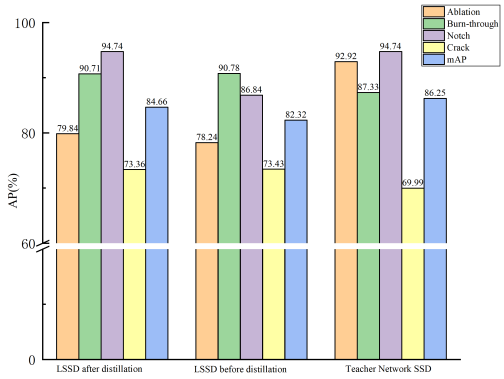
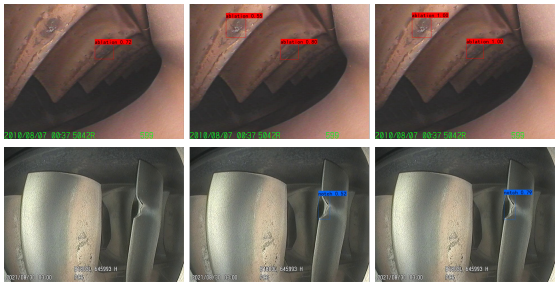


Fig. 4. Experimental results of distillation based on intermediate layer feature knowledge



(a) LSSD before distillation (b) LSSD after distillation (c) SSD

Fig. 5. Comparison of detection results

3.4. Comparative experiments

In order to verify the performance of the lightweight object detection model LSSD in borescope detection, we compare it with Faster RCNN, RetinaNet, SSD, YOLOv4, and typical lightweight object detection models, such as YOLOv3-tiny, YOLOv4-tiny, YOLOv5-s, LSSD with Mobilenetv3-small and Mobilenetv3-large backbone. The results are shown in Table 4. Compared with Faster RCNN, SSD, RetinaNet, YOLOv4, LSSD has significantly reduced parameter and

Table 3. Comparison of time performance and performance with different superpixel methods

Number of anchor boxes	Anchor box size on feature map 38 × 24	Anchor box size on feature map 19 × 12	Anchor box size on feature map 10×6	mAP /%	Params /M	MACs /G
6	[8,7] [17,22]	[47,30] [39,49]	[69,43] [105,60]	82.32	4.99	3.54
8	[17,15] [23,18]	[56,48][50,76] [81,72]	[124,79][108,129] [185,111]	78.42	5.03	3.56
10	[8,7][17,22] [22,18][40,59]	[47,30][39,49] [46,80][61,52]	[84,82][123,133] [134,53]	71.29	5.07	3.59
12	[8,7][17,22] [22,18][40,59]	[85,32][58,50] [64,68][46,96]	[86,84][127,77] [189,68][104,131]	70.76	5.11	3.62
14	[8,7][17,22] [22,18][40,59] [43,32]	[59,50][45,88] [75,66][88,90] [139,54]	[124,80][103,134] [176,83][163,147] -	72.28	5.15	3.66
16	[8,7][18,14] [23,18][41,36] [40,60][58,49]	[44,98][83,59] [137,53][87,83] [86,107]	[124,81][179,78] [102,140][123,136] [165,148]	74.31	5.19	3.69
18	[8,7][18,14] [23,19][39,36] [40,59][86,33]	[59,49][46,91] [69,67][90,82] [159,60][90,110]	[125,80][99,143] [117,129][167,ca101] [149,141][205,163]	69.19	5.23	3.70

computational complexity, and its detection speed also meets the requirements of real-time detection; compared with other lightweight models, the LSSD model has the advantage of high detection accuracy.

Meanwhile, we have deployed our model on different devices. Among different CPUs, the inference time is 200ms on i5-12600KF and 395ms on i7-6700K. Among different GPUs, the inference time is 10ms on RTX3060 and 55ms on GTX1070. It can be seen that LSSD performs well and is suitable for deployment in practical environments.

4. Conclusions

We propose a lightweight model LSSD to address the issues of a large number of parameters, high computational complexity, poor real-time detection, and difficulty in deploying on mobile detection devices in the field of borescope. This model achieves real-time detection of blade damage in borescope videos, and the following conclusions can be drawn:

1. The introduction of the inverse residual structure has lightweight the SSD backbone network and the K-means clustering algorithm is adopted to optimize the size and number of anchor boxes, which can reduce the proposed LSSD model's parameter and computational complexity, and improve its detection speed.

2. The addition of the improved feature fusion module CA-FPN not only deeply integrates the semantic information of the deep network with the edge information of the shallow network, enhances the model's detection ability for small and medium-sized targets, but also promotes the model to more accurately identify and locate blade damage. Embedding a small-scale damage detection enhancement module consisting of multi-scale dilated convolution and residual connections in the small-scale damage detection branch not only enhances the perception ability of shallow feature maps to global features, but also preserves important information from input feature maps. The average detection accuracy of the proposed model is improved by 1.2%.

3. The knowledge distillation strategy of intermediate layer features is used to distill and train the LSSD model. The average detection accuracy was improved by 2.34%, the ablation detection accuracy was improved by 1.6%, and the notch detection accuracy was improved by 7.9%.

4. The params of the LSSD model is 4.99M, the MACs is 3.541G, the detection speed is 32 FPS, and the average detection accuracy on the blade damage image test set is 84.66%. Compared with the SSD model, the LSSD model has reduced the number of parameters and computation by 79.3% and 88.42% respectively, and the detection speed has increased by 16 frames per second.

References

[1] K. G. Harding, J. Gu, L. Tao, G. Song, and J. Han. "Blade counting tool with a 3D borescope for turbine applications". In: *Interferometry XVII: Advanced Applications*. ***9204***. SPIE. 2014, 143–152.

Table 4. Comparison of Results between LSSD Model and Other Object Detection Models

Models	mAP@0.5/%	Params/M	MACs/G	Speed/FPS
Faster RCNN	90.1	136.771	184.909	8
RetinaNet	91.40	36.392	78.988	13
SSD	86.25	24.146	30.577	16
YOLOv3-tiny	81.54	8.674	6.446	32
YOLOv4	84.01	63.959	70.984	20
YOLOv5-s	90.53	7.7074	8.255	32
YOLOv4-tiny	84.38	5.883	8.095	33
LSSD+Mobilenetv3-small	69.86	7.617	16.705	30
LSSD+Mobilenetv3-large	78.63	11.243	21.216	29
LSSD(ours)	84.66	4.99	3.541	32

[2] Z. Yuan. "Borescope inspection for HPT blade of CFM56-7B engine". In: *IOP Conference Series: Materials Science and Engineering*. ***382***. IOP Publishing. 2018, 032028.

[3] D. Zhang, N. Zeng, and L. Lin. "Detection of blades damages in aero engine". In: *2020 Chinese Automation Congress (CAC)*. IEEE. 2020, 6129–6134.

[4] J. Aust and D. Pons, (2019) *"Taxonomy of gas turbine blade defects"* **Aerospace *6***(5): 58.

[5] R. Mishra, J. Thomas, K. Srinivasan, V. Nandi, and R. R. Bhatt, (2017) *"Failure analysis of an un-cooled turbine blade in an aero gas turbine engine"* **Engineering Failure Analysis *79***: 836–844.

[6] J. Aust and D. Pons, (2019) *"Bowtie methodology for risk analysis of visual borescope inspection during aircraft engine maintenance"* **Aerospace *6***(10): 110.

[7] J. Aust and D. Pons, (2020) *"A systematic methodology for developing bowtie in risk assessment: application to borescope inspection"* **Aerospace *7***(7): 86.

[8] M. Li-Yong, H. Chun-Sheng, and H. Yu-Qing. "Defect detection of the irregular turbine blades based on edge pixel direction information". In: *2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC)*. IEEE. 2018, 165–170.

[9] S. Shao and W. Lin. "Compressor Blade Fault Diagnosis Based on Image Processing". In: *2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE)*. IEEE. 2018, 467–471.

[10] C. Y. Wong, P. Seshadri, and G. T. Parks. "Automatic borescope damage assessments for gas turbine blades via deep learning". In: *AIAA Scitech 2021 Forum*. 2021, 1488.

[11] C. Cai, Y. Ziyan, and S. Lizhong, (2023) *"Aeroengine damage detection method based on improved YOLOv4"* **Modern Manufacturing Engineering**: 99–108. DOI: 10.16731/j.cnki.1671-3133.2023.02.014.

[12] H. Yuhao, C. Xueguo, L. Xinliang, J. Haokun, and W. Jingqiu, (2024) *"Real time detection of aircraft engine blade damage based on SW-YOLO model"* **Journal of Propulsion Technology *45***: 197–203. DOI: 10.13675/j.cnki.tjjs.2302058.

[13] X. Li, W. Wang, L. Sun, B. Hu, L. Zhu, and J. Zhang, (2022) *"Deep learning-based defects detection of certain aero-engine blades and vanes with DDSC-YOLOv5s"* **Scientific Reports *12***(1): 13067.

[14] H. Shang, C. Sun, J. Liu, X. Chen, and R. Yan, (2022) *"Deep learning-based borescope image processing for aero-engine blade in-situ damage detection"* **Aerospace Science and Technology *123***: 107473.

[15] W. Yongchao, L. Tao, and D. Yi, (2022) *"Research on Intelligent Detection Method of Engine Hole Detection Defects Based on Raspberry PI"* **Modern Computer *28***(8): 100–103+108. DOI: 10.3969/j.issn.1007-1423.2022.08.018.

[16] A. G. Howard, (2017) *"Mobilenets: Efficient convolutional neural networks for mobile vision applications"* **arXiv preprint arXiv:1704.04861**:

[17] X. Zhang, X. Zhou, M. Lin, and J. Sun. "Shufflenet: An extremely efficient convolutional neural network for mobile devices". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 6848–6856.

[18] Y. Liu, Z. Wang, X. Wu, F. Fang, and A. S. Saqlain, (2022) *"Cloud-edge-end cooperative detection of wind turbine blade surface damage based on lightweight deep learning network"* **IEEE Internet Computing *27***(1): 43–51.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. "Ssd: Single shot multibox detector". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, 21–37.

[20] S. Ren, K. He, R. Girshick, and J. Sun, (2016) *"Faster R-CNN: Towards real-time object detection with region proposal networks"* **IEEE transactions on pattern analysis and machine intelligence** ***39***(6): 1137–1149.

[21] J. Redmon. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, 4510–4520.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 2117–2125.

[24] Q. Hou, D. Zhou, and J. Feng. "Coordinate attention for efficient mobile network design". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, 13713–13722.

[25] G. Hinton, (2015) *"Distilling the Knowledge in a Neural Network"* **arXiv preprint arXiv:1503.02531**:

[26] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, (2014) *"Fitnets: Hints for thin deep nets"* **arXiv preprint arXiv:1412.6550**: