

Tiny Defect Detection in High-Resolution Aero-Engine Blade Images via a Coarse-to-Fine Framework

Dawei Li^{ID}, Yida Li^{ID}, Qian Xie^{ID}, Yuxiang Wu^{ID}, Zhenghao Yu^{ID}, and Jun Wang^{ID}

Abstract—This article studies the problem of aero-engine blade surface defect detection in large images. The effective method for aero-engine blade surface inspection for this real application is currently lacking since most defects are relatively small. Therefore, the task of aero-engine blade surface defect detection is mainly implemented by experienced operators, which is subjective and time-consuming. Moreover, it is hard to fit the requirements of higher accuracy and efficiency manually. Therefore, an effective and efficient method for aero-engine blade surface defect detection is demanded. To achieve this, we propose a vision-based framework in this study to detect defects in a coarse-to-fine manner. First, the captured raw images are with a high resolution of 2448×2048 to ensure the accuracy of defect detection. The raw images are then cropped into smaller regions and fed into our deep convolution neural network (DCNN) to learn features with high representation. Next, the coarse classifier module is proposed to filter most background regions out. Finally, the defects are located and classified by a fine detector module in the defective images, which are selected by the coarse classifier module. Instead of directly applying a detector, our coarse-to-fine framework can effectively save computation and improve accuracy. In addition, the coarse-to-fine framework can be trained in an end-to-end manner. Compared with classical methods for object detection, our method also achieves state-of-the-art performance for aero-engine blade surface defect detection in terms of accuracy and efficiency. Furthermore, our framework has been applied for practical application in many aero-engine blade production lines.

Index Terms—Aero-engine blade, coarse classifier, coarse-to-fine framework, deep learning (DL), fine detector, tiny defect detection.

I. INTRODUCTION

AERO-ENGINE blade inspection is an important and essential part of the production process since a defective blade can force the airplane engine to shut down, which is

Manuscript received December 16, 2020; revised January 31, 2021; accepted February 14, 2021. Date of publication February 24, 2021; date of current version March 23, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1707504 and Grant 2020YFB2010702, in part by the National Natural Science Foundation of China under Grant 61772267, in part by the Aeronautical Science Foundation of China under Grant 2019ZE052008, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190016. The Associate Editor coordinating the review process was Eduardo Cabal-Yepez. (*Corresponding author: Jun Wang*)

Dawei Li, Yida Li, Qian Xie, Yuxiang Wu, and Jun Wang are with the College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: dawei@nuaa.edu.cn; nuaa.yld@gmail.com; qianxie@nuaa.edu.cn; yuxiangwu@nuaa.edu.cn; wjun@nuaa.edu.cn).

Zhenghao Yu is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: yvzhenghao@outlook.com).

Digital Object Identifier 10.1109/TIM.2021.3062175

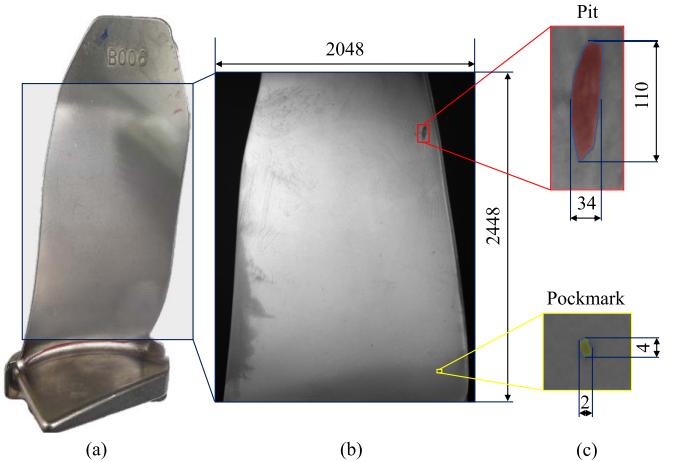


Fig. 1. Example of an aero-engine blade that contains defects. (a) Aero-engine blade model. (b) Captured large image of the aero-engine plane, which is with the resolution of 2448×2048 . (c) Two defects (pit and pockmark) of an aero-engine blade surface, which are within the different scales. Specifically, the pockmark only occupies several pixels in the large image.

destructive and dangerous. Therefore, high-quality production of produced aero-engine blades is required. However, defects frequently occur during the process of aero-engine blade production, which are relatively small. Although most defects are less than 1 mm, the threats to airplane safety are serious. Nevertheless, the automatic and effective method for aero-engine blade surface defect detection is still lacking.

Traditionally, to inspect the aero-engine blade automatically, some nondestructive techniques have developed in recent years, including radiographic testing [1], ultrasonic testing [2], magnetic particle testing [3], and eddy current testing [4]. These methods have achieved good performance for aero-engine blade internal flaw detection, while they are not suitable for defect detection of aero-engine blade surfaces. Most aero-engine blade defects are relatively small and so are easily missed by these methods, as shown in Fig. 1. Therefore, defects on aero-engine blade surfaces are mainly detected by experienced inspectors, whose intensive work gives subjective results with low efficiency. The tiny defects are also easily missed by experienced inspectors.

In recent years, the techniques [5]–[10] based on computer vision (CV) and deep learning (DL) have been developed rapidly, which are applied in many fields for defect detection [11]–[14], [15]. These methods have achieved outstanding performance for metal surface defect detection [16]–[18].

Therefore, this is a good direction for aero-engine blade surface defect detection by vision-based methods since the aero-engine blade is also belonging to metal production.

However, vision-based methods [5]–[7] still have not been applied for aero-engine blade defect detection with good performance. This is because defects on aero-engine blade surfaces are relatively small, as shown in Fig. 1. The tiny defects are easily missed by learning-based methods [5], [6] [19]. Due to a series of downsampling operations, such as pooling, the tiny defect regions are easily filtered out based on the classical convolutional neural networks (CNNs) [20]–[22]. In addition, tiny defects are also easily blurred by convolution and pooling operations with network depth increasing. Moreover, some defects within different scales can be missed by classical methods [20]–[22]. In summary, the problems of aero-engine blade surface defect detection include the following.

- 1) The defects of aero-engine blade surfaces are relatively small, which are easily missed by the classical learning-based methods, since the tiny defect features are blurred by the downsampling operations.
- 2) The resolution of the captured raw images is high, while the single tiny defect only occupies several pixels in the raw image, which is hard to detect in the whole image region.
- 3) Scale variation in defect detection of aero-engine blade surfaces frequently occurs, which is also difficult to detect by the classical methods.

In this article, we propose a new framework for aero-engine blade surface defect detection, which is in a coarse-to-fine manner. First, we use charge-coupled device (CCD) cameras with high resolution to capture aero-engine blade surface image data, which are in the size of 2448×2048 . Second, the collected images are cropped into smaller ones for further analysis. Third, to detect defects in different scales, we propose a new backbone network to learn robust features for defect prediction, which applies different size kernels for feature learning. Moreover, to avoid feature blur, a few pooling operations are applied in the backbone network. Forth, since most image regions belong to the background, we propose a coarse classification module to filter out the nondefective regions. Finally, the extracted defective patches are fed into the defect fine detector module to locate and classify the aero-engine blade surface defects. In addition, we design a new loss function for the framework training in an end-to-end manner. Our framework has been applied to many aero-engine blade production lines.

In short, our contributions are summarized as follows.

- 1) We design a coarse-to-fine framework to detect tiny defects of aero-engine blade surfaces in large images automatically, which is accurate and efficient.
- 2) We propose a novel backbone network for image feature learning, which is constructed by different size convolutional kernels and less pooling operations.
- 3) We propose a coarse classifier module to filter out the nondefective regions, which can save computation and reduce the interferences caused by a complex background.

- 4) To achieve robust training in an end-to-end manner, we design a new loss function for our framework, which is named coarse-to-fine loss (CTFL).

II. RELATED WORK

With the rapid development of the aviation industry, there have been some measurements [23]–[25] proposed for engine blades' inspection. Zhang *et al.* [26] proposed a method to identify the status of the blades by using acoustic emission (AE) monitoring. This method was used to predict the degree of crack propagation and residual fatigue life based on the AE energy that was obtained. Schlobohm *et al.* [27] presented four complementary nondestructive measurement techniques for material characterization and damage detection of turbine blades, including macroscopic fringe projection with inverse fringe projection algorithms, robot-guided microscale fringe projection, high-frequency eddy current, and pulsed high-frequency induction thermography. Burghardt *et al.* [28] presented a method for robot-assisted geometrical inspection of an aircraft engine turbine stator segment. They employed the ABB IRB 1600 robot, equipped with the Atos Core 3-D scanner and interfaced with the Atos Professional software suite to inspect geometrical characteristics. These methods performed well for engine blade inspection and measurement. However, these methods cannot fit the process of aero-engine blade production. Moreover, the aero-engine blade surface defects cannot be detected by these methods. To address these issues, we propose to use vision-based methods to detect defects of aero-engine blade surfaces during the process of production since it has high accuracy and efficiency.

In recent years, the techniques [5]–[9], [29], [30] of image processing and DL have developed rapidly. The tasks for image classification [31]–[34], object detection [35]–[38], and image segmentation [39]–[41] have become the most popular research areas, which have been commonly applied to many practical applications [14], [42]–[44]. However, during the process of aero-engine blade production, the product surface inspection measurements are still primarily manual, which is time-consuming and subjective. The tiny defects easily miss detection by these methods. Recently, some learning-based algorithms [45], [46] have been proposed to detect defects of the aero-engine blade surfaces. Kim and Lee [45] introduced an algorithm to recognize damage to engine blades from video-scope images, which consists of image preprocessing, CNN, and GUI components. This method has achieved good performance for damage recognition of engine blades. However, it could not detect tiny defects with high accuracy since this method treats the task of defect detection as general image classification, which would result in feature missing of small defects after a series of convolution and pooling operations. Therefore, it was not suitable for aero-engine blade surface defect detection during the process of product manufacturing since most defects are within small regions. Chen *et al.* [46] proposed a new network named a feature weighting network (FWNet) to solve the problem of defect scale variation and improve defect detection accuracy. This network achieved good performance for aircraft engine defect detection. However, the algorithm was computationally complex for defect

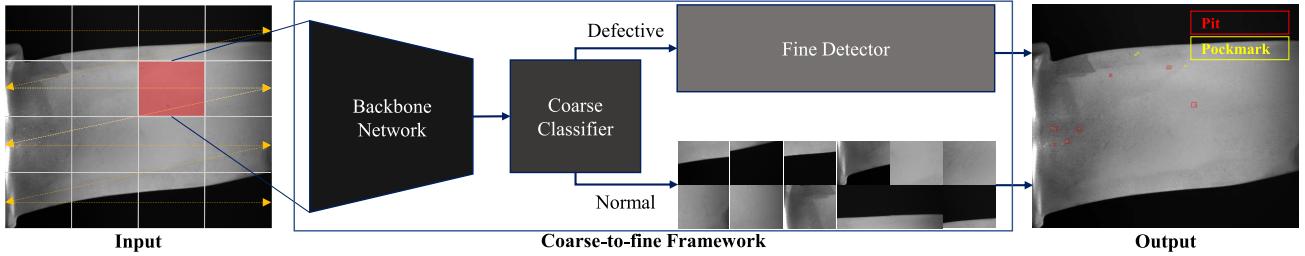


Fig. 2. Pipeline of our method for aero-engine blade surface defect detection. First, the single raw image is cropped into sixteen patches. Second, the cropped patches are fed into the backbone network for feature learning successively, following the order of the yellow arrows. Third, the cropped patches are classified as defective ones or normal by the coarse classifier module. Finally, the extracted defective ones are fed into the object fine detector module for further prediction of defect localization and fine classification. The final output of our framework is the single image with labeled defects.

detection in large images. To this end, we propose a coarse-to-fine framework to detect aero-engine blade surface defects in large images with high accuracy and efficiency.

In total, an efficient and effective method for aero-engine blade surface defect detection is still lacking since most defect regions are relatively small in the captured large images. To this end, we propose a vision-based method to detect aero-engine blade surface defects, which is in a coarse-to-fine manner. Our method achieves good performance in terms of accuracy and efficiency.

III. METHOD

In this study, we propose a learning-based method to implement defect detection of aero-engine blade surfaces in a coarse-to-fine manner, which is accurate and efficient, as shown in Fig. 2. The detailed description of our framework for aero-engine blade defect detection in images is illustrated as follows. It consists of the *Backbone Network* for feature learning, the *Coarse Classifier* for image extraction containing defects, and the *Fine Detector* for fault classification and location. In addition, the *Loss Function* is designed for end-to-end training.

A. Coarse-to-Fine Framework

Since the resolution of captured images is high, it would result in lacking memory and high computational complexity via DL. To address this issue, the raw images with the resolution of 2448×2048 are cropped into smaller ones, which are in the size of 612×512 . Specifically, the single raw image can be cropped into sixteen smaller ones that are suitable for the processes of training and testing.

The most collected image regions belong to the background, which would lead to complex calculation, if they are fed into the deep network. However, the state-of-the-art frameworks [5], [47]–[49] for object detection just make a prediction for the whole image, which is computationally complex. To achieve fast defect detection without a decrease in accuracy of the aero-engine blade surfaces, we construct a novel end-to-end framework, which is in the manner of coarse-to-fine, as shown in Fig. 3.

First, the raw image is cropped into smaller ones and fed into the backbone network for feature learning. However, most aero-engine blade surface defects are small, which easily

miss detection by the classical learning-based methods, since many downsampling operations are inserted, such as pooling. To avoid the feature blur of tiny defects by the convolution and pooling operations, we propose a new backbone network with fewer pooling layers than several classical networks [20], [21], [50]. Moreover, different size kernels are adopted to learn the defect features with multiscales. Concatenation and convolution operations are then applied to fuse the feature maps learned by multiscale kernels.

Second, the learned last layer feature map is fed into the coarse classifier to judge whether or not the image contains defects at the image level, which is based on the softmax model. The function of this module is to reduce the computational complexity since most background regions are filtered out. Furthermore, if the nondefective ones are classified as defective ones, they would be fed into the fine detector module for deeper analysis. Therefore, the last part, namely, the fine detector module, can also filter the missing classified ones out. This part is defined as a coarse module since the defect analysis lacks fine classification and localization.

Finally, after the process of coarse classification, the defective ones are selected for further prediction, including defect localization and fine classification. The input of this module is the last fused feature map of the backbone network, which is in a high resolution. Due to the background filer by the coarse classifier, the computation of this module is dramatically decreased. Moreover, the accuracy will obviously increase since most interference of background is filtered out by the coarse classifier module. The final output of our network is the image with located and classified defects, which are labeled in the images. This new network achieves defect detection in the aero-engine blade surface images with high accuracy and efficiency.

B. Backbone Network

Most aero-engine blade surface defects are small in size, only occupying several or dozens of pixels in large images. Moreover, due to the pooling operation, the feature map resolution is dramatically decreased by the classical backbone networks [20], [21], [50], which leads to poor performance for tiny defect detection. To this end, we design a novel backbone network with a few downsampling operations, such as pooling, since the tiny defects are easily filtered out by pooling and

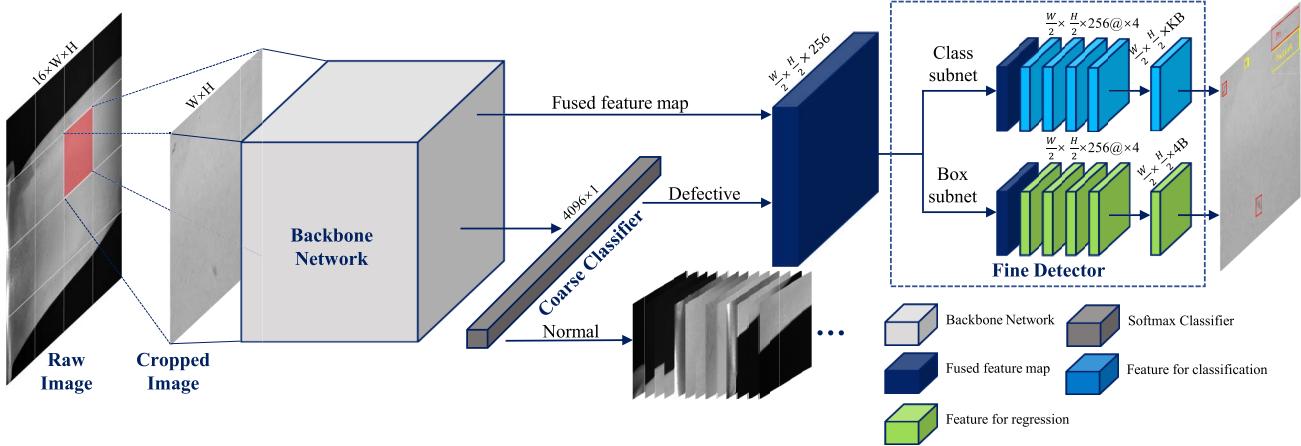


Fig. 3. Architecture of our framework for defect detection of aero-engine blade surfaces in large images, including the backbone network module, the coarse classifier module, and the object fine detector module. Specifically, the backbone network module is to learn feature maps with high representation; the coarse classifier is to filter the background images out; and the object fine detector is to achieve defect localization and fine classification.

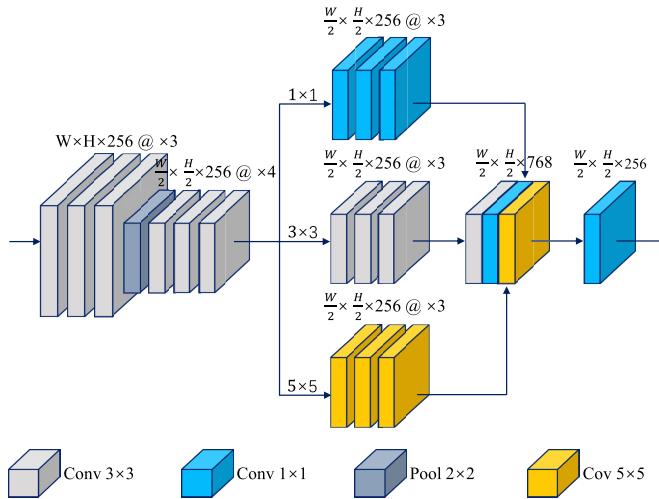


Fig. 4. Backbone network of our framework. To detect tiny defects from the cropped regions with high accuracy, a few downsampling operations are applied in the network since the tiny defect regions are easily filtered out and blurred by convolution and pooling operations. Moreover, we adopt different kernels for image feature learning since most aero-engine blade defects are within different scales. The feature maps learned by different kernels are then fused for further prediction.

convolution operations. Furthermore, some defects are within different scales. To enhance the accuracy for defect detection in different scales, we propose to use three different kernels for feature map learning since the different size kernels are sensitive to the different scale defects. The feature maps learned by different kernels are then concatenated. Next, a convolution operation is followed to compress the concatenated feature map into the fixed depth, as shown in Fig. 4.

In detail, the first three convolutional layers maintain the resolution with the original input image. The convolution kernels are within the size of 3×3 . A 2×2 pooling operation is then executed to filter the useless information out and reduce computation. After the pooling operation, the feature map resolution is dropped into the half size of the original

input image. Next, another three convolution operations are followed, which are in the size of 3×3 . After that, three parallel convolution operation branches are executed, which are implemented by different size kernels of 1×1 , 3×3 , and 5×5 . Specifically, each branch contains three convolutional layers. Afterward, the feature maps learned by different kernels are concatenated. Finally, the 1×1 convolution operation is implemented to compress the concatenated feature map into the fixed depth of 256. The processes can be modeled as

$$X_f = \psi_f[\chi_i(F_i)], \quad (1)$$

where χ_i means the transformation function of each source feature map before being concatenated, F_i indicates each feature map before being transformed, i denotes each branch label, ψ_f is the feature map fusion function, and X_f is the fused feature map. The detailed description of the backbone network is concluded as in Algorithm 1.

C. Coarse Classifier

The resolution of captured raw images of aero-engine blade surfaces is extremely high. However, most image regions belong to the background. Therefore, it would result in high computational cost if all the image regions are fed into the fine detector module directly. To this end, we propose to add a coarse classifier module to filter the most background regions out. This module can save about 70% of the computation for defect detection since more than 90% of regions belong to the background. In detail, the classifier is constructed by a fully connected layer and the softmax function [51]. Especially, the softmax function is used as the last activation function to normalize the output of a network to a probability distribution over predicted classes. Specifically, the fully connected layer is within the resolution of 4096×1 . The softmax function decides whether the cropped region belongs to defective ones or those that are normal. If the cropped region belongs to the defective ones, it would be fed into the object fine detector module for further prediction, including localization and fine classification. Moreover, if the cropped region does not contain

Algorithm 1 Backbone Network

Input: An image I with resolution of $W \times H$.

Step 1: Image I is fed into backbone network and is processed by three layer convolution kernels with the resolution of 3×3 . The feature map $Fconv_1$ is generated.

Step 2: $Fconv_1$ is then downsampled by a 2×2 pooling kernel. The feature map $Fpool_1$ is generated.

Step 3: $Fpool_1$ is processed by next three layer convolution kernels with the resolution of 3×3 . The feature map $Fconv_2$ is generated.

Step 4: The feature map $Fconv_2$ is then processed by three types of convolution kernels, which are with the resolution of 1×1 , 3×3 , and 5×5 . And the feature maps of $Fconv_3_1$, $Fconv_3_3$, and $Fconv_3_5$ are generated.

Step 5: Concatenate the feature maps of $Fconv_3_1$, $Fconv_3_3$, and $Fconv_3_5$ as a new feature map of $Fconcat_1$.

Step 6: $Fconcat_1$ is fed into a 1×1 convolution layer to compress the depth into 256. And the last output of the backbone network is a feature map $Fconv_4$ with the resolution of $\frac{W}{2} \times \frac{H}{2} \times 256$.

Output: The feature map $Fconv_4$.

defects, it would be classified as normal. The next patch is then fed into the backbone network for another prediction.

D. Fine Detector

After the process of coarse classification, the regions classified as defective ones are then fed into the object fine detector module for further prediction. To locate and classify aero-engine blade surface defects with high accuracy, we propose an object fine detector module embedded in our framework, including a classification subnet and a box regression subnet. Specifically, the two subnets are responsible for defect fine classification and localization respectively, as shown in the object fine detector module of Fig. 3.

1) Classification Subnet: This classification subnet predicts the defect probability for each of the B anchor boxes for K classes. The subnet is a small fully connected network (FCN) attached to the fused feature map, which is learned by the backbone network module. The FCN includes five 3×3 convolution layers. The first four convolution layers have 256 filters and all followed by ReLU activations; a 3×3 convolution layer with KB filters is followed. Then ReLU activations are attached to output the KB predictions for each location. Especially, the ReLU activation is applied to improve network representation, which is defined as

$$f(x) = \max(0, x), \quad (2)$$

where x is the input to a neuron. Moreover, a vector with the length of K is generated for each anchor B . Therefore, the depth of the last layer feature map of the class subnet is set as KB to predict the classification of each anchor. In this article, $B = 9$ and $K = 5$ are applied in the experiments.

2) Box Regression Subnet: Compared with the classification subnet, another small subnet, namely Box Regression Subnet, is also attached to the fused feature map for regressing the offset from each anchor box to the corresponding ground-truth box. The structure of the designed box regression subnet is identical to the classification subnet. However, the last output feature map is in the depth of $4B$. For each anchor box location, the four outputs predict the offset between the anchor box and the nearby ground-truth box. Especially, the ground-truth box is selected by the highest intersection over union (IoU) score with the anchor box. Moreover, the classification subnet and the box regression subnet share a common input feature map, namely, the fused feature map. In addition, the two subnet structures are usually the same while using separate parameters.

E. CTFL Definition

To achieve robust training for our end-to-end framework, we construct a CTFL function in this study, which includes two parts. The first part of CTFL is the loss for image-level classification (L_{cls}). The definition for L_{cls} is described as

$$L_{cls}(q_i, q_i^*) = \sum_i (-q_i^* \log(q_i) - (1 - q_i^*) \log(1 - q_i)) \quad (3)$$

where i is the number of cropped regions in a training batch. q_i^* is the ground-truth label. If the label of the cropped region is positive, it would be set to 1; if the label of the cropped region is negative, it would be set to 0. Moreover, q_i is the predicted label. The other part of CTFL is the loss for defect localization and fine classification. However, since the defect regions are relatively small, it would lead to extreme foreground-background class imbalance. This may cause overfitting during the training process of the object fine detector. To this end, the Focal Loss (FL) [49] is applied for the detector training, which is defined as

$$FL_P = \sum_j (-(1 - P_j)^\gamma \log(P_j)) \quad (4)$$

where j is the number of defect samples in a training batch, γ is the tunable focusing parameter, and $\gamma \geq 0$; P is defined as

$$P = \begin{cases} p & y = 1 \\ 1 - p & \text{otherwise} \end{cases} \quad (5)$$

where $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$.

Since most cropped regions belong to negative ones, it is inevitable to lead to overfitting, which is caused by foreground-background class imbalance. To this end, the balancing parameter λ is added to our loss function

$$CTFL(q_i, P_j) = FL_P + \lambda L_{cls}(q_i, q_i^*) \quad (6)$$

where $\lambda \in [0, 1]$. Specifically, the higher λ can improve the accuracy of coarse classification, while the accuracy for defect fine detection would decrease.

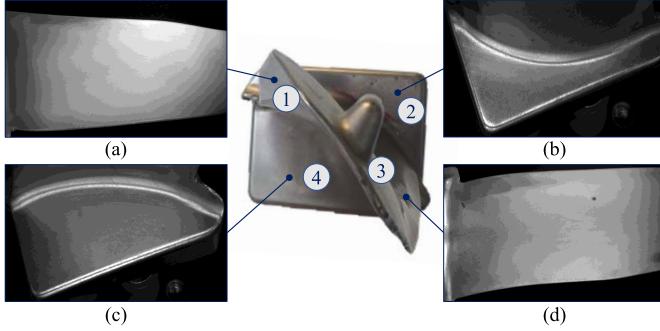


Fig. 5. Example of collected blade images, which are captured from four different surface regions. (a)–(d) Collected images of surfaces 1, 2, 3, and 4 respectively.

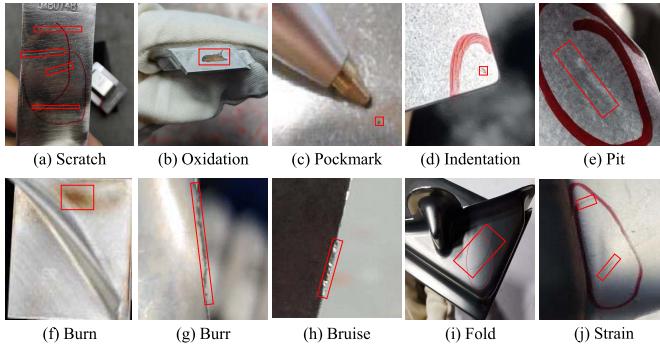


Fig. 6. Ten types of defects on the aero-engine blade surface. Especially, the defects that frequently occur are scratch, pockmark, pit, fold, and strain. These defects are taken into model evaluation in this study. (a) Scratch. (b) Oxidation. (c) Pockmark. (d) Indentation. (e) Pit. (f) Burn. (g) Burr. (h) Bruise. (i) Fold. (j) Strain.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setting

1) *Data Set Preparation*: Note that all defects are visually labeled by experienced experts. Fig. 5 depicts the raw images with high resolution, which are captured from the four regions of a single aero-engine blade surface. First, the collected raw images are cropped into smaller size ones, which are appropriate for network training. The cropped regions are then classified as defective ones and normal ones by the image level. This pattern of image classification is to fit the module training of the coarse classifier, which is to filter the background regions out. Next, the defective ones are extracted for fine labeling by defect-level manually. As shown in Fig. 6, there are ten types of defects during the process of aero-engine blade production. However, since most defects rarely occur, five types of defects are selected for model evaluation in this study. Specifically, these five defects are including pit, pockmark, scratch, fold, and strain, which occupy more than 80% of all types of the defects, as shown in Fig. 5. Finally, the labeled images are classified as the training set, validation set, and testing set.

Specifically, the raw images are captured from 500 aero-engine blade surfaces, including 2000 large images with a high resolution of 2448×2048 . After the process of image cropping, there are 32 000 smaller ones. To avoid overfitting

TABLE I
DATA ON THE NUMBER OF FRAMEWORK TRAINING,
VALIDATION, AND TESTING

Data	Raw image	Training	Validation	Testing
Defective	1,488	9,940	1,242	1,242
Normal	512	9,940	1,242	1,242
Total	2,000	19,880	2,484	2,484

caused by imbalanced training data, the number of smaller normal ones applied for network training is equal to the defective ones. Therefore, there are 24 848 cropped images, including 12 424 normal ones and 12 424 defective ones, which are selected for training, validation, and testing. As presented in Table I, 80% of the data are randomly selected for training; 10% of the data are randomly selected for validation, and 10% of the data are selected for testing.

2) *Evaluations*: To test the performance of our method for aero-engine blade surfaces defect detection, some commonly used indicators are introduced, including accuracy (A), precision (P), recall (R), and F1 score ($F1$). Accuracy measures the ratio of correctly predicted defects to the total detected defects. Precision represents the ratio of detection that is true positive, whereas recall measures the ratio of correctly predicted defects to the total positives. $F1$ score is applied to weight the average of precision and recall, which takes the false positives and false negatives into account. In detail, these measurements are given as

$$A = \frac{TP + TN}{TP + FP + FN + TN} \quad (7)$$

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$

$$F1 = \frac{2 * (Recall * Precision)}{Recall + Precision} \quad (10)$$

where TP is the true positive, FN is the false negative, FP is the false positive, and TN is the true negative. True positive represents the correctly predicted positives; false negative represents the falsely predicted negatives; false positive represents the falsely predicted positives; and true negative represents the correctly predicted negatives.

3) *Training Scheme*: In this study, we implement our framework training in an end-to-end manner, which is based on the TensorFlow API [52]. First, we set the momentum optimizer with the rate of 0.9 and a weight decay of 0.0001. The learning rate is initially set as 0.001 and decreased to 0.0005 after 40k iterations. In addition, the stochastic gradient descent (SGD) [53] is applied for model weight optimization. Moreover, the hyperparameter setting of $CTFL$ is relatively important since it can directly affect the convergence directly. In detail, γ is set as 2, and λ is set as 0.5 of our loss function during the process of model training. Furthermore, we initialize our backbone network by ImageNet [54] pre-trained weights. The experimental environment is described as follows: CPU clocked at 3.70 GHz and an NVIDIA GeForce RTX 2080Ti graphic processing unit (GPU) card with

TABLE II
COMPARED RESULTS FOR BACKBONE NETWORK FEATURES LEARNING, WHICH ARE BASED ON THE BASELINE OF FASTER R-CNN AND OUR COARSE-TO-FINE FRAMEWORK

Baseline	Backbone	Testing Set	Resolution	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Faster R-CNN []	VGG16 [20]	200	2448×2048	72.1	73.6	78.9	76.2
	VGG19 [20]	200	2448×2048	73.5	74.7	79.7	76.8
	ResNet-50 [21]	200	2448×2048	75.9	77.4	81.8	79.5
	ResNet-101 [21]	200	2448×2048	76.2	78.1	82.0	80.0
	GoogleNet [51]	200	2448×2048	78.6	79.8	83.1	81.4
	Ours	200	2448×2048	87.4	89.9	95.5	92.6
Coarse-to-fine Framework	VGG16 [20]	200	2448×2048	75.6	77.2	80.3	78.7
	VGG19 [20]	200	2448×2048	77.0	78.6	80.6	79.6
	ResNet-50 [21]	200	2448×2048	79.2	80.9	82.0	81.4
	ResNet-101 [21]	200	2448×2048	80.3	81.3	82.7	82.0
	GoogleNet [51]	200	2448×2048	81.9	82.7	83.6	83.1
	Ours	200	2448×2048	93.5	94.8	96.1	95.4

TABLE III
EVALUATION RESULTS FOR COARSE CLASSIFIER MODULE. THE COARSE CLASSIFIER MODULE CAN IMPROVE THE ACCURACY AND EFFICIENCY FOR AERO-ENGINE BLADE SURFACE TINY DEFECT DETECTION

Baseline	Coarse Classifier	Testing Set	Resolution	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Time (ms)
Our Backbone Network	-	200	2448×2048	91.0	92.7	95.8	94.2	186
+ RetinaNet [50]	✓	200	2448×2048	91.7	94.2	95.3	94.7	59

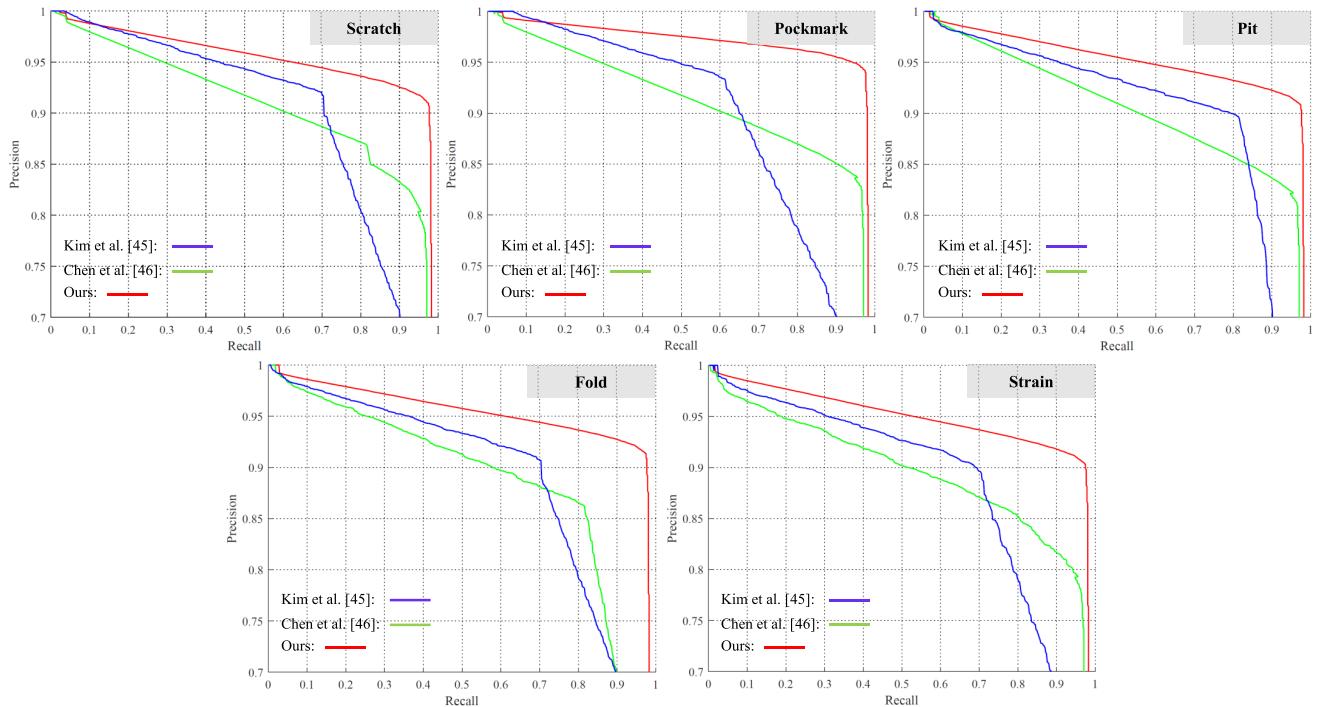


Fig. 7. Precision–recall curves for the defect detection results of scratch, pockmark, pit, fold, and strain. Specifically, we compare our method with another two state-of-the-art methods for aero-engine blade defect detection. Our method gets prior performance for defect detection for aero-engine blade defect detection.

11-GB memory. Moreover, the experiments are implemented by PYTHON with the Pytorch tool.

B. Ablation Study and Analysis

1) Comparison Study for Backbone Network: For better understanding our backbone network for aero-engine blade

surface image feature learning, we construct a series of ablation studies on our data set using Faster R-CNN [5] and our coarse-to-fine framework, separately.

In this study, the classical backbone networks for object detection are taken into account for comparison, including VGG16, VGG19 [20], ResNet-50, ResNet-101 [21], and GoogleNet [50], which are mostly applied for the tasks of

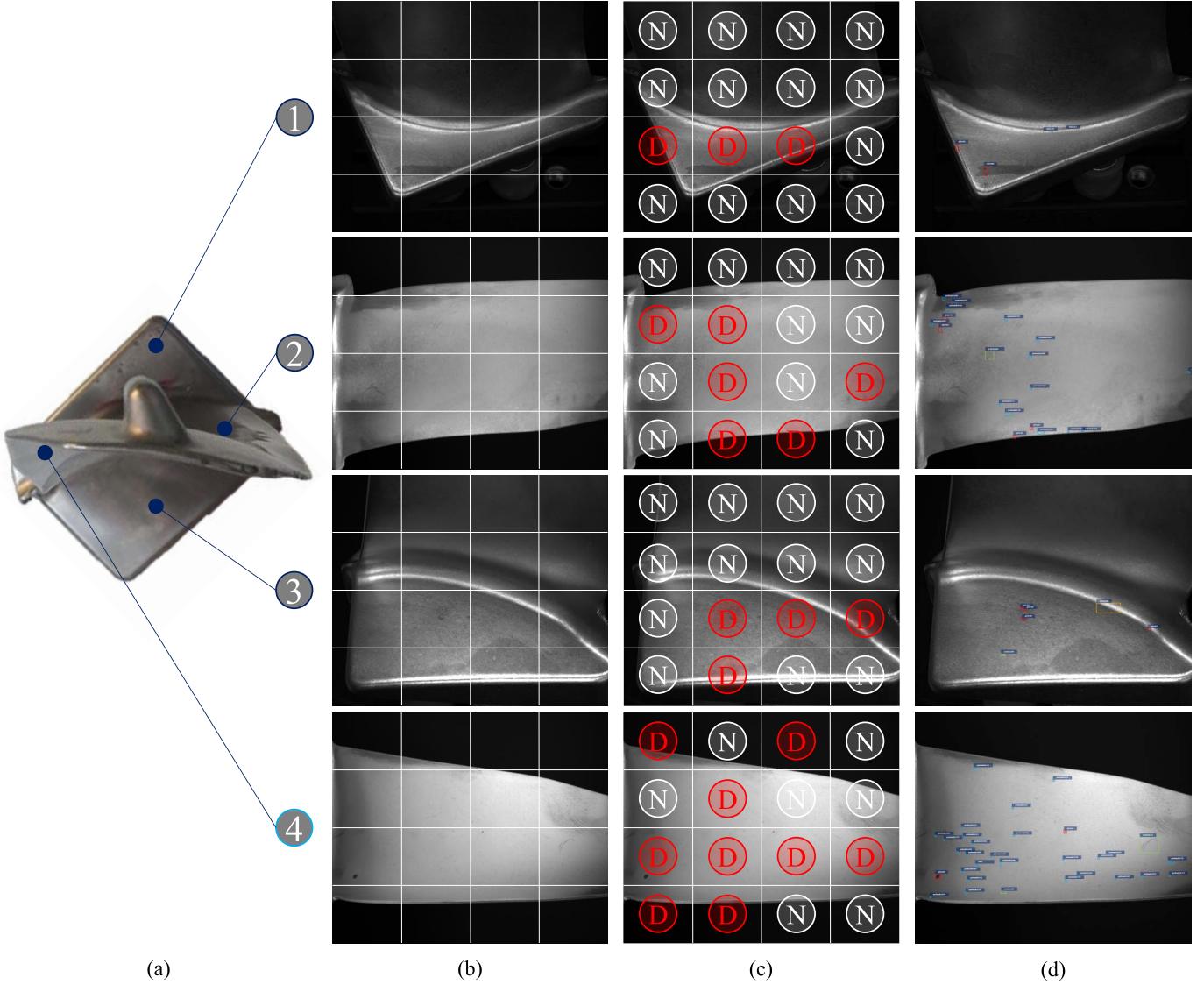


Fig. 8. Example of detected results of an aero-engine blade with intermediate results. (a) Aero-engine blade with four regions to be inspected. (b) Divided regions from the captured raw images. (c) Coarse classified results for defect detection. N represents the regions that are classified as normal; D represents the regions that are classified as defective ones. (d) Results with defect localization and fine classification of the four aero-engine blade surface regions.

TABLE IV
COMPARED RESULTS OF DETECTED DEFECTS IN THE CROPPED AERO-ENGINE BLADE SURFACE DEFECTIVE IMAGES. WE COMPARE WITH THE CLASSICAL OBJECT DETECTION METHODS, INCLUDING FASTER R-CNN [5], YOLOV4 [47], CASCADE R-CNN [48], AND RETIANET [49]

Backbone	Testing Set	Resolution	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Faster R-CNN [5]	2,484	612×512	72.1	73.6	78.9	76.2
YOLOv4 [48]	2,484	612×512	75.9	78.7	80.3	79.5
Cascade R-CNN [49]	2,484	612×512	77.0	79.1	82.9	81.0
RetinaNet [50]	2,484	612×512	77.5	78.5	79.6	79.0
Ours	2,484	612×512	93.5	94.8	96.1	95.4

object detection. This comparison is to verify the efficiency of our backbone network for tiny defect detection in large images. As shown in Table II, we implement our comparison based on the baseline of Faster R-CNN and our coarse-to-fine framework. Our backbone network achieves the best performance for aero-engine blade surface defect detection, in terms

of accuracy, precision, recall, and *F1* score. Specifically, the accuracy, precision, recall, and *F1* score of our method are 93.5%, 94.8%, 96.1%, and 95.4%, respectively, based on the Faster R-CNN, which outperforms the classical methods. In addition, the performance of our backbone network is also superior to the several classical networks that are compared

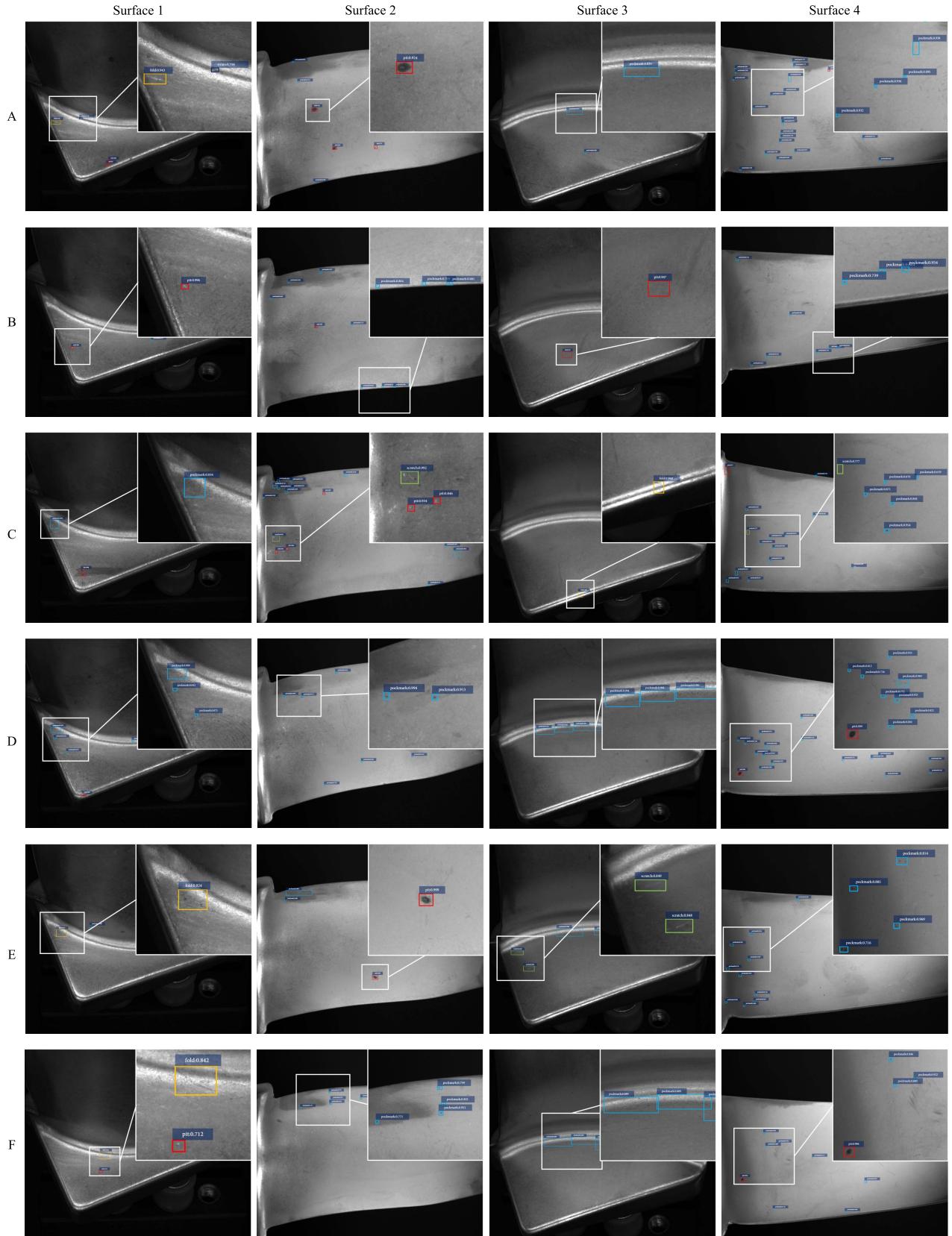


Fig. 9. Some detected results of blade surface defects. Specifically, the blades A, B, C, D, E, and F with detected defects are presented. Each blade is inspected with four surfaces 1, 2, 3, and 4.

based on our coarse-to-fine framework. To the best of our knowledge, this is because we apply fewer downsampling operations, such as pooling, which can blur tiny defects.

However, there are many downsampling operations in the classical backbone networks, which would lead to features missing of tiny defects. Moreover, the operation of applying

TABLE V

COMPARED RESULTS OF DETECTED DEFECTS IN THE CROPPED AERO-ENGINE BLADE SURFACE DEFECTIVE IMAGES. WE COMPARE WITH THE CLASSICAL OBJECT DETECTION METHODS, INCLUDING FASTER R-CNN, YOLOV4, CASCADE R-CNN, AND RETIANNET

Methods	Testing Set Number	Resolution	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Kim et al. [46]	200	2448×2048	78.4	79.6	81.8	80.7
Chen et al. [47]	200	2448×2048	83.9	85.5	86.2	85.8
Ours	200	2448×2048	93.5	94.8	96.1	95.4

different kernels for feature learning also helps further enhance the representation of learned feature maps. Therefore, we get the state-of-the-art performance in terms of defect feature learning in aero-engine blade surface large images.

2) *Comparison Study for Coarse Classifier*: To filter the most background regions out, the coarse classifier module is proposed and inserted into our framework. This operation can save the most computation and improve the accuracy for aero-engine blade surface tiny defect detection. To verify the effectiveness and efficiency of this proposed module, we compare our framework with RetinaNet [49]. In addition, the comparison is based on the baseline of our backbone network and fine detector module. We evaluate the performance of our model from two aspects: accuracy and efficiency. During the training process of RetinaNet, the feature map is fed into the fine detector module directly. As presented in Table III, our model achieves the superior results for tiny defect detection in terms of accuracy (91.7%), precision (94.2%), *F1* (94.7%) score, and average detection time (59 ms) for a single large image, which are tested on the 200 testing data set. Our method is three times faster than RetinaNet since most image regions belong to the background, which is filtered out by the coarse classifier module. However, the coarse classifier can lead to a recall decrease (95.8%->95.3%) since several images containing defects are filtered out by this module.

3) *Comparison Study for Fine Detector*: To achieve the tiny defect detection with high accuracy, we propose a fine detector module. The fine detector module is used to locate and classify defects in the defective images, which are selected by the coarse classifier module. To test its performance for defect detection in the cropped defective images, we compare our method with Faster R-CNN [5], YOLOv4 [47], Cascade R-CNN [48], and RetianNet [49]. To make a fair comparison, these methods are in the same training and testing scheme. Indeed, these methods are all based on our feature learning network. Furthermore, the testing process is implemented based on the testing set.

As shown in Table IV, the detected results by our method are superior to these classical object detectors in terms of accuracy (93.5%), precision (94.8%), recall (96.1%), and *F1* score (95.4%). This is because we use fewer downsampling operations in the fine detector module. Moreover, the deeper convolution layers help to improve the representation of the learned feature map for defect classification and regression prediction.

C. Comparison With State-of-the-Art Methods

To further test the performance of our framework, we compare our method with several state-of-the-art

algorithms [45], [46] for aero-engine blade surface defect detection. Chen *et al.* [46] proposed a new network named FWNet for aircraft engine defect detection, which is designed based on CNNs and a feature pyramid. Kim and Lee [45] also proposed an inspection method for engine blades using CNNs and image processing, while the defect fine detection module is lacking in their network. These two algorithms have achieved outstanding performance for blade inspection during the maintenance phase. However, blade defect detection during the process of production is lacking. Therefore, we compare our method with [45], and [46] in this study.

As shown in Table V and Fig. 7, the defect detection results implemented in [45] and [46] are relatively good. Hence, our method achieves better results for aero-engine blade surface defect detection in terms of accuracy and efficiency, including accuracy (93.5%), precision (94.8%), recall (96.1%), and *F1* score (95.4%). The reason is that the defects are relatively small, which easily missed detection by the two classical methods for blade inspection. Moreover, the captured raw images are in a high resolution of 2448 × 2048. This can lead to high computation using the two methods directly since most background regions are fed into the deep networks. To solve these issues, the proposed coarse classifier filters the most background regions out. This operation can save most computation and further improve the accuracy during the process of defect fine detection. Therefore, we achieve superior performance for aero-engine blade surface defect detection in large images. Some detected results are shown in Figs. 8 and 9.

V. CONCLUSION AND FUTURE WORK

In this article, we propose a coarse-to-fine framework to detect defects in the large images of aero-engine blade surfaces. Our method achieves state-of-the-art performance in terms of accuracy, precision, recall, *F1* score, and efficiency. The reasons for achieving relatively good performance are mainly including three parts. First, we propose a backbone with fewer downsampling operations during the process of feature learning. This operation helps improve the accuracy of tiny defect detection efficiently. The coarse classifier module is then proposed to filter the most background regions out. This module can save the most computation and further enhance the defect detection accuracy since most image regions belong to the background. Moreover, the fine detector module is proposed to locate and classify defects in the defective regions with high accuracy. In total, our method achieves good performance for defect detection of aero-engine blade surface in large images, which has been applied to many aero-engine blade production lines.

Though our method has achieved good performance for aero-engine blade surface defect detection in large images, there are five types of defects that have not been taken into consideration for the model training. Since the other five types of defects rarely occur, the number of image data containing these defects is relatively small. To detect all types of aero-engine blade surface defects with high accuracy, we will try to collect more image data. Moreover, we will try the few-shot learning-based methods [56]–[58] to avoid the problems of small data set and sample number imbalance. Above all, we will try our best to further enhance the accuracy and efficiency for aero-engine blade surface defect detection. In addition, detecting all types of defects automatically is another task that we will pursue. Moreover, the problem of defect splitting will be taken into consideration. Furthermore, we will try to apply our method to gas turbine blade defect detection for further applications.

REFERENCES

- [1] W. K. Wong, S. H. Ng, and K. Xu, “A statistical investigation and optimization of an industrial radiography inspection process for aero-engine components,” *Qual. Rel. Eng. Int.*, vol. 22, no. 3, pp. 321–334, 2006.
- [2] V. Ageeva, T. Stratoudaki, M. Clark, and M. G. Somekh, “Integrative solution for *in-situ* ultrasonic inspection of aero-engine blades using endoscopic cheap optical transducers (CHOTs),” in *Proc. 5th Int. Symp. NDT Aerosp. (AeroNDT)*, Singapore, Nov. 2013.
- [3] F. Zou, “Review of aero-engine defect detection technology,” in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Jun. 2020, pp. 1524–1527.
- [4] B. Sasi, B. Rao, and T. Jayakumar, “Dual-frequency eddy current non-destructive detection of fatigue cracks in compressor discs of aero engines,” *Defence Sci. J.*, vol. 54, no. 4, p. 563, 2004.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [6] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [7] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
- [8] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
- [10] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
- [11] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, “Automatic detection and classification of sewer defects via hierarchical deep learning,” *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1836–1847, Oct. 2019.
- [12] X. Xu, Y. Lei, and F. Yang, “Railway subgrade defect automatic recognition method based on improved faster R-CNN,” *Sci. Program.*, vol. 2018, Jun. 2018, Art. no. 4832972.
- [13] D. Li *et al.*, “Automatic defect detection of metro tunnel surfaces using a vision-based inspection system,” *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101206.
- [14] Y. He, K. Song, Q. Meng, and Y. Yan, “An end-to-end steel surface defect detection approach via fusing multiple hierarchical features,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, Apr. 2020.
- [15] S. Mei, H. Yang, and Z. Yin, “An unsupervised-learning-based approach for automated defect inspection on textured surfaces,” *IEEE Trans. Instrum. Meas.*, vol. 67, no. 6, pp. 1266–1277, Jun. 2018.
- [16] Q. Jiang, D. Tan, Y. Li, S. Ji, C. Cai, and Q. Zheng, “Object detection and classification of metal polishing shaft surface defects based on convolutional neural network deep learning,” *Appl. Sci.*, vol. 10, no. 1, p. 87, Dec. 2019.
- [17] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, “Automated defect inspection system for metal surfaces based on deep learning and data augmentation,” *J. Manuf. Syst.*, vol. 55, pp. 317–324, Apr. 2020.
- [18] I. Katsamenis, E. Protopapadakis, A. Doulamis, N. Doulamis, and A. Voulodimos, “Pixel-level corrosion detection on metal constructions by fusion of deep learning semantic and contour segmentation,” 2020, *arXiv:2008.05204*. [Online]. Available: <http://arxiv.org/abs/2008.05204>
- [19] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 379–387.
- [20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” 2016, *arXiv:1602.07261*. [Online]. Available: <http://arxiv.org/abs/1602.07261>
- [22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size,” 2016, *arXiv:1602.07360*. [Online]. Available: <http://arxiv.org/abs/1602.07360>
- [23] G. Bolu, A. Gachagan, G. Pierce, and G. Harvey, “Reliable thermosonic inspection of aero engine turbine blades,” *Insight-Non-Destructive Test. Condition Monitor*, vol. 52, no. 9, pp. 488–493, Sep. 2010.
- [24] D. Szybicki, A. Burghardt, P. Gierlak, and K. Kurc, “Robot-assisted quality inspection of turbojet engine blades,” in *Proc. Int. Workshop Model. Social Media*. Cham, Switzerland: Springer, 2018, pp. 337–350.
- [25] X.-J. Lin, C.-W. Shan, Z.-Q. Wang, and Y.-Y. Shi, “Measurement techniques of coordinate measuring machine for blade surface of aero-engine,” *Comput. Integrat. Manuf. Syst.*, vol. 18, no. 1, pp. 125–131, 2012.
- [26] Z. Zhang, G. Yang, and K. Hu, “Prediction of fatigue crack growth in gas turbine engine blades using acoustic emission,” *Sensors*, vol. 18, no. 5, p. 1321, Apr. 2018.
- [27] J. Schlobohm *et al.*, “Advanced characterization techniques for turbine blade wear and damage,” *Procedia CIRP*, vol. 59, pp. 83–88, Jan. 2017.
- [28] A. Burghardt, K. Kurc, D. Szybicki, M. Muszyńska, and T. Szczęch, “Robot-operated inspection of aircraft engine turbine rotor guide vane segment geometry,” *Tehnicki Vjesnik-Tech. Gazette*, vol. 24, no. 2, pp. 345–348, 2017.
- [29] J.-D. Tournier *et al.*, “MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation,” *NeuroImage*, vol. 202, Nov. 2019, Art. no. 116137.
- [30] M. Nixon and A. Aguado, *Feature Extraction and Image Processing for Computer Vision*. New York, NY, USA: Academic, 2019.
- [31] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.
- [32] I. Zeki Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, “Billion-scale semi-supervised learning for image classification,” 2019, *arXiv:1905.00546*. [Online]. Available: <http://arxiv.org/abs/1905.00546>
- [33] Y. Zhu *et al.*, “Multi-representation adaptation network for cross-domain image classification,” *Neural Netw.*, vol. 119, pp. 214–221, Nov. 2019.
- [34] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, “Fundus image classification using VGG-19 architecture with PCA and SVD,” *Symmetry*, vol. 11, no. 1, p. 1, Dec. 2018.
- [35] G. Ghiasi, T.-Y. Lin, and Q. V. Le, “NAS-FPN: Learning scalable feature pyramid architecture for object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7036–7045.
- [36] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.
- [37] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
- [38] J. Wang, L. Luo, W. Ye, and S. Zhu, “A defect-detection method of split pins in the catenary fastening devices of high-speed railway based on deep learning,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9517–9525, Dec. 2020.
- [39] S. Ghosh, N. Das, I. Das, and U. Maulik, “Understanding deep learning techniques for image segmentation,” *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–35, Sep. 2019.
- [40] C. Liu *et al.*, “Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–92.

- [41] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.
- [42] Q. Xie, D. Li, Z. Yu, J. Zhou, and J. Wang, "Detecting trees in street images via deep learning with attention module," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5395–5406, Aug. 2020.
- [43] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, Mar. 2020.
- [44] D. He, K. Xu, and P. Zhou, "Defect detection of hot rolled steels with a new object detection framework called classification priority network," *Comput. Ind. Eng.*, vol. 128, pp. 290–297, Feb. 2019.
- [45] Y.-H. Kim and J.-R. Lee, "Videoscope-based inspection of turbofan engine blades using convolutional neural networks and image processing," *Structural Health Monitor.*, vol. 18, nos. 5–6, pp. 2020–2039, Nov. 2019.
- [46] L. Chen, L. Zou, C. Fan, and Y. Liu, "Feature weighting network for aircraft engine defect detection," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 18, no. 3, May 2020, Art. no. 2050012.
- [47] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: <http://arxiv.org/abs/2004.10934>
- [48] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
- [49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [50] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [51] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 2016, *arXiv:1611.01144*. [Online]. Available: <http://arxiv.org/abs/1611.01144>
- [52] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: <http://arxiv.org/abs/1603.04467>
- [53] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Cham, Switzerland: Springer, 2010, pp. 177–186.
- [54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [55] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8420–8429.
- [56] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4013–4022.
- [57] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13846–13855.



Dawei Li received the bachelor's degree in machine design & manufacturing and their automation from the Anhui University of Technology (AHUT), Ma'anshan, China, in 2016. He is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China.

His research interests include computer vision and deep learning.



Yida Li received the bachelor's degree in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2019, where he is currently pursuing the master's degree.

His research interests include computer vision and robotics.



Qian Xie received the bachelor's degree in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2015, where he is currently pursuing the Ph.D. degree.

His research interests include computer vision, robotics, and machine learning.



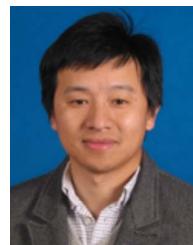
Yuxiang Wu received the bachelor's degree in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2020, where he is currently pursuing the master's degree.

His research interests include computer vision and deep learning.



Zhenghao Yu received the bachelor's degree in electrical engineering and automation from Henan Polytechnic University Jiaozuo, China, in 2016. He is currently pursuing the master's degree with the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, from 2017.

His research interests include computer vision and machine learning.



Jun Wang received the bachelor's and Ph.D. degrees in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2002 and 2007, respectively.

From 2008 to 2010, he conducted research as a Post-Doctoral Scholar at the University of California at Davis, Davis, CA, USA, and the University of Wisconsin, Madison, WI, USA. From 2010 to 2013, he worked as a Senior Research Engineer at Leica Geosystems, Duluth, GA, USA. In 2013, he paid an academic visit to the Department of Mathematics, Harvard University, Cambridge, MA, USA. He is currently a Professor with NUAA. His research interests include geometry processing and geometric modeling.