Journal of Manufacturing Systems 72 (2024) 59–73

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Uncertainty-driven trustworthy defect detection for high-resolution powder bed images in selective laser melting

Zhibin Zhao [a,b], Weilin Liu [a,b], Jiaxin Ren [a,b], Chenxi Wang [a,b,*], Yixuan He [c], Xingwu Zhang [a,b], Xuefeng Chen [a,b]

[a] National Key Lab of Aerospace Power System and Plasma Technology, Xi'an Jiaotong University, Xi'an 710049, PR China
[b] School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, PR China
[c] Xi'an Kongtian Electromechanical Intelligent Manufacturing Co., Ltd, PR China

ARTICLE INFO

Keywords:
High-resolution powder bed image
Trustworthy defect detection
Uncertainty estimation
Selective laser melting

ABSTRACT

Selective laser melting (SLM) is known as one of the most promising metal additive manufacturing technologies, and how to ensure its consistent quality is still a main challenge, which urgently needs to be addressed. The metal powder spreading quality, as the first stage of SLM, has a direct impact on the subsequent forming and it is necessary to be monitored. However, existing deep learning-based methods for monitoring powder spreading quality often suffer from the problem of unreliability. The high-resolution property of powder bed images is often ignored, and the predicted results are not well evaluated. To address the above issues and achieve trustworthy defect detection, this paper proposes an uncertainty-driven trustworthy defect detection method for high-resolution powder bed images in SLM. A super resolution module based on ISDNet is first proposed to refine the upsampling process. Checkpoint ensemble is adopted for better achieving model uncertainty estimation only requiring a single training process, feature reuse helps reduce the computational load, and temperature scaling is used to further calibrate each ensemble member to get a more precise uncertainty. Besides, we design an uncertainty-driven model improvement method to further improve defect detection performance for regions with high uncertainty. Experiments demonstrate the effectiveness of our proposed method, and our work achieves a trustworthy defect detection for high-resolution powder bed images in SLM.

1. Introduction

Selective laser melting (SLM) is known as one of the most promising metal additive manufacturing (MAM) technologies [1]. It utilizes the high-energy density laser beam as heat source to melt the mental powder selectively according to the planned path layer by layer [2]. Benefited from the tiny laser spot size, typically ranged from 20 μm to 100 μm, SLM can manufacture high-density components with high forming precision, and is suitable for the manufacture of special complex structures, such as thin walls, complex curved surfaces, and spatial lattices. Compared with the traditional manufacturing process, this technology can greatly save the processing cycle, avoid waste of materials, and reduce mold costs [3]. Therefore, SLM has been widely used in aerospace, high-end equipment, and industrial design fields [4]. However, how to ensure the reliability of component quality and the repeatability of manufacturing has gradually been becoming the biggest challenge for SLM, which has been considered as one of the main obstacles to the development and batch industrial application of SLM and other MAM technologies [5]. During the metal powder spreading process, some factors, like the vibration of the recoater blade and the insufficient amount of powder, will lead to the unevenness of powder bed. If defects of the powder spreading are not detected, evaluated, and disposed in time, it will inevitably cause hidden defects in the forming component, even leading to processing failure [6]. It is unacceptable to use the components with internal damages and the processing failure will cause huge economic losses, especially for the components with a processing cycle of several months. Therefore, it is quite important to monitor the powder spreading quality during the SLM process to manufacture high-performance mental components.

Machine vision has been widely adopted to monitor the powder spreading quality. It is worth mentioning that the resolution of these images collected from the camera is relatively large due to the fact that the width of the power bed is generally large and the size of the defects is usually small. The higher the resolution, the more defect information

* Corresponding author at: National Key Lab of Aerospace Power System and Plasma Technology, Xi'an Jiaotong University, Xi'an 710049, PR China.
E-mail address: wangchenxi@xjtu.edu.cn (C. Wang).

https://doi.org/10.1016/j.jmsy.2023.11.006
Received 30 June 2023; Received in revised form 10 October 2023; Accepted 13 November 2023
Available online 27 November 2023

can be obtained. However, high resolution will result in the burden of computational time for feature extraction. Generally, there are three schemes based on machine vision to carry out the powder spreading defect detection. One scheme is to obtain the height distribution of the powder bed surface by means of active illumination and coherent interference. Neef et al. [7] applied low coherence interferometric imaging technology to detect the flatness of the powder bed in SLM process, which scans the powder bed with a measuring laser beam, measures the optical path difference between the reflected light and the reference light with a spectrometer, and then compensates the deviation caused by the angular deflection to obtain the height distribution of different scanning points. This technology can effectively detect the ups and downs of the powder bed and can identify grooves with a depth of 50 μm on the powder bed. Zhang et al. [8] proposed an in-situ fringe projection technology to reconstruct the deformation stripes captured by the camera to obtain the three-dimensional shape of the object, and realize the detailed measurement of the surface topography of the powder layer and molten metal surface. DePond et al. [9] adopted high-speed spectral domain optical coherence tomography along the laser light path to measure the powder height of each layer after melting to explore the mechanism of defect formation, thereby improving process parameters. Cao et al. [10] applied the structured light technology to measure the flatness and contour of the additive manufacturing process online and analyzed the flatness of the forming working surface to estimate the quality of powder spreading. Based on the height distribution of powder bed surface, the quality of powder spreading can be estimated intuitively and effectively. However, these methods usually require active lighting and special image acquisition equipment, which is expensive and not conducive to the arrangement in airtight chamber.

Additionally, some research analyzing the grayscale texture of the powder bed picture to detect powder spreading defects has been conducted due to the lower computational complexity and good explanation. This scheme depends on the height distribution of powder bed surface essentially because different heights will lead to different gray values on the powder bed picture under certain lighting conditions. Therefore, the grayscale texture of the powder bed image can be analyzed by image processing techniques, such as morphological filtering, to distinguish the location and scope of powder spreading defects. Craeghs et al. [11] extracted some stripes perpendicular to the powder spreading direction on powder bed grayscale images and compared the average value with a reasonable grayscale range to effectively identify shallow ravines along the moving direction of the recoater blade caused by wear and local damages. Jacobsmuhlen et al. [12] performed threshold processing on grayscale images and realized the effective extraction of both position and area of the super-elevation according to the characteristics of the bright area generated by the specular reflection of the super-elevation. Abdelrahman et al. [13] extracted the area corresponding to the cross-section of the part from powder bed images and superimposed it to form a three-dimensional part model, which can well reflect the three-dimensional space position of the powder bed anomaly. Lin et al. [14] used threshold segmentation to extract the stripe and cladding defects from powder bed images after eliminating the effect of light, and then multilayer perceptron and support vector machine were used to perform classification. However, this process relies on many manually designed feature extractors, requires professional knowledge and complex parameter tuning process, and is aimed at specific applications with relatively worse generalization ability and robustness.

Recently, due to the powerful representation ability, deep learning can automatically extract features from massive data and has attracted increasingly attention in powder spreading defect detection. This scheme directly uses deep learning models to classify or locate the defects from the original images. Scime et al. [15] built a powder bed image dataset to detect and classify defects using an unsupervised machine learning algorithm. They later combined the AlexNet, Multi-scale convolutional neural network (CNN) and transfer learning to carry out the powder spreading anomaly detection [16]. Based on previous work, they further introduced an additional UNet to return pixel-wise segmentation results for defect location [17]. The algorithm was validated on six different devices, covering three molding techniques including laser fusion, binder jetting, and electron beam fusion. Chen et al. [18] proposed a two-stage CNN to detect and segment the powder spreading defects. They first adopted EfficientNet B7 to classify the images which contained defects and then applied Mask R-CNN to locate the defect area. Mehta et al. [19] applied the UNet to divide the whole powder bed into three kinds of area: powder, part, and defect. Besides, they utilized federated learning to alleviate the constraints of data availability and data privacy. Fischer et al. [20] designed a recoater-based line sensor to acquire images of powder bed with 6 μm/pixel resolution and applied Xception to classify image patches into different defects. The powder spreading defect detection based on deep learning can mine the advanced semantic features of various defects, which eliminates the cumbersome manual extraction. However, as mentioned above, powder bed images captured by the industrial camera usually features high resolution and existing methods mainly focus on regular resolution images instead of the feasibility of larger scale input due to the limitation of computational speed. Above methods either downsample high-resolution powder bed images to fit deep learning models, leading to loss of image details, or divide powder bed images into patches for prediction and then stitch them back together, leading to lose the contextual semantics of the patch edges. How to directly and precisely classify and locate the defects from the high-resolution powder bed images while maintaining low computational time is still an unsolved problem.

However, the deployment of deep learning in real industrial scenarios is often hampered by the inherently "black box" property, leading to the fact that users often cannot trust the results from deep learning models. That is, it is often unknowable what the outcome of a single prediction is and why it is the case, which can seriously affect the security of the decision and cannot achieve a trustworthy result. Therefore, it is urgent to address whether and how the prediction results of a deep learning model are trustworthy. A trustworthy deep learning model should be capable of giving accurate predictions, evaluate the credibility of predictions additionally, and warn experts in the decision loop to handle anomalies when the predictions have high uncertainty. However, deep learning models often only give point predictions, fail to evaluate the uncertainty, and are often overconfident or diffident in the prediction results, which makes their results unreliable and not trustworthy.

- To address above issues, this paper proposes an uncertainty-driven trustworthy defect detection method for high-resolution powder bed images in SLM. Our main contributions can be summarized as follow:To achieve defect segmentation of high-resolution powder bed images with high accuracy, a super resolution module to refine the upsampling process based on ISDNet is proposed, which well balances the performance and computational burden for defect detection of high-resolution powder bed images and can obtain a finer-grained powder spreading defect segmentation mask.
- To precisely evaluate the credibility of prediction results, a model ensemble approach is designed to achieve model uncertainty estimation, which only requires a single training process. Temperature scaling is used to calibrate each ensemble member to eliminate the cognitive bias and get a more precise uncertainty.
- To further utilize uncertainty for trustworthy defect detection, an uncertainty-driven model improvement method is proposed for regions with high uncertainty to enhance powder spreading defect detection performance.
- Powder spreading defect image dataset, including super-elevation, incompletion, hopping, streaking, and lattice, is collected and labeled during the metal powder spreading process of real

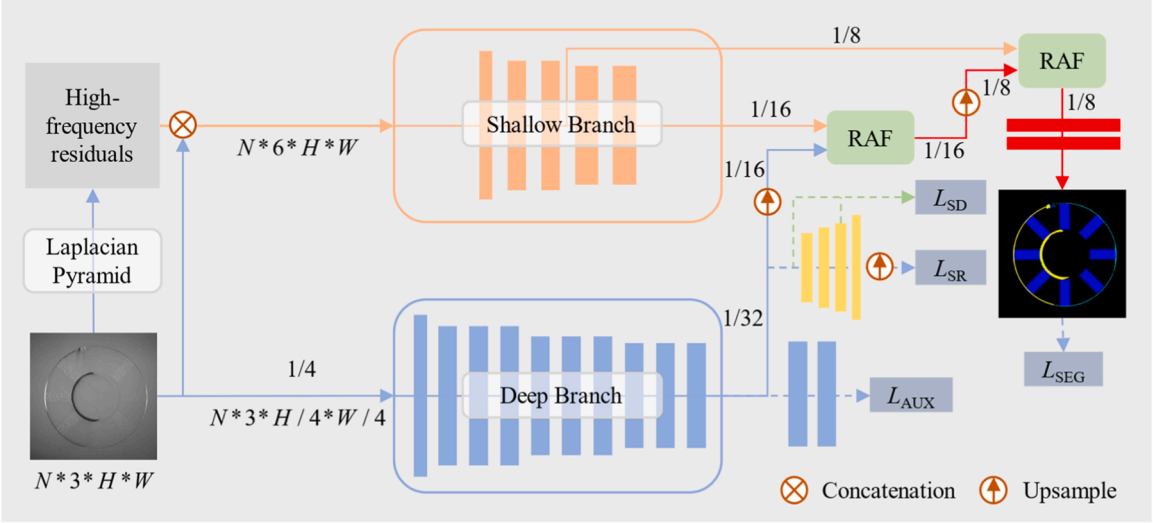

Fig. 1. The pipeline of ISDNet [31].

components manufacture by SLM. Then, the established dataset is used to verify the performance of the proposed method.

The organization of this paper is summarized as follows: Section 2 presents related research, including semantic segmentation methods for high-resolution images and uncertainty evaluation methods in deep learning. Our proposed approaches are described in Section 3 and verified their effectiveness in Section4. In Section 5, our work is summarized.

2. Preliminary

2.1. High-resolution image segmentation

Higher resolution means more details but leads to huge burden for the computation and memory, which needs special designs to deal with. Some works based on the combination of global features and local features has been conducted [21–24]. In the way, high-resolution images are downsampled to extract global features while patches are chosen from original resolution images by some strategies to extract local features. Local features are utilized to compensate for the loss of detail information caused by downsampling at the original position. Besides, refining the segmentation results by a cascaded structure with reusing intermediate features helps deal with the high-resolution image segmentation problem [25–28]. In this case, a fine-grained mask is obtained by fusing the coarse mask with other intermediate features in each iteration and then feeding to the next iteration. The intermediate features in each iteration are considered to contain some information, which the output mask does not have and is beneficial to obtain a fine-grained mask. Last, the fusion of low-level and high-level features has also attracted attention in this field [29–31]. Low-level features have higher resolution and retain more image texture information, while high-level features contain advanced semantic information and facilitate pattern recognition. The effective fusion of these two kinds of features can meet the demand for high-resolution image segmentation to obtain fine-grained masks. In summary, the first two schemes either need to be carried out in two stages or need to repeat the same operation many times, and such serial operations are time-consuming. The last scheme, which fuses low-level features with high-level features, can well work with the special design of effective feature fusion modules.

Among these methods, ISDNet[31] has proved its ability to balance accuracy and speed, outperforming Unet on the DeepGlobe dataset, and the whole structure is shown in Fig. 1. It is attributed to the effective fusion of high-level features and low-level features. ISDNet employs DeepLabv3 with ResNet18 as the deep branch to extract high-level features and the lightweight model called Short-Term Dense Concatenate (STDC) [32], in which only the first four stages are used, to extract low-level features. Original images are downsampled by four times as the input of the deep branch while high-frequency residuals, which are computed from full-resolution images through Laplacian pyramid, are concatenated with full-resolution images as the input to the shallow branch. Assume that the original image has the shape of $N*3*H*W$, the shape of deep branch' input is $N*3*H/4*W/4$, while the shape of shallow branch' input is $N*6*H*W$. For better feature fusion, a relation-aware feature fusion module (RAF) is designed as follows:

Let $F_d \in \mathbb{R}^{C\times H_d\times W_d}$ and $F_s \in \mathbb{R}^{C\times H_s\times W_s}$ donate high-level features and low-level features, respectively. Global average pooling (GAP) is performed separately for each of them, and each obtains a one-dimensional feature vector with the length equal to the number of channels. Afterwards, a multilayer perceptron (MLP) is used to obtain the respective channel attention weights, donated as $w_d^{\mathrm{CA}} \in \mathbb{R}^{1\times C_d}$ and $w_s^{\mathrm{CA}} \in \mathbb{R}^{1\times C_s}$. This process is described as:

$$w^{\mathrm{CA}} = \mathrm{MLP}\big(\mathrm{GAP}(F)\big) \tag{1}$$

The channel attention weights of high-level features and low-level features are orderly divided into$k$groups with length$r$to rearranged into feature matrices, donated as$\mathrm{G}_d \in \mathbb{R}^{k\times r}and\mathrm{G}_s \in \mathbb{R}^{k\times r}$. For easy understanding, assume$\mathrm{G}_d = [\alpha_1, \alpha_2, \ldots, \alpha_k]^Tand\mathrm{G}_s = [\beta_1, \beta_2, \ldots, \beta_k]^T$, where$\alpha_kand\beta_k$are both column vector with length$r$. The feature relationship matrix is obtained as:

$$\mathrm{R} = \mathrm{G}_s\mathrm{G}_d^{\mathrm{T}}. \tag{2}$$

For any element$R_{i,j}$in matrix$R$, it is calculated as follow:

$$R_{i,j} = \beta_i^T\alpha_j = <\beta_i, \alpha_j>, \tag{3}$$

where$<\beta_i, \alpha_j>$represents the inner product of$\beta_i$andα_j, which describes the relevance between two vectors. Through Formula (2), each group of channel attention weights from high-level features can make inner product with all groups of channel attention weights from low-level features, respectively. In this way, the feature relationship between low-level features and high-level features is established.

To further explore the feature interactions between high-level features and low-level features, the feature relationship matrix is flattened into a one-dimensional vector. The feature interaction reorganization is performed using a MLP, and the feature interaction weight is obtained as:

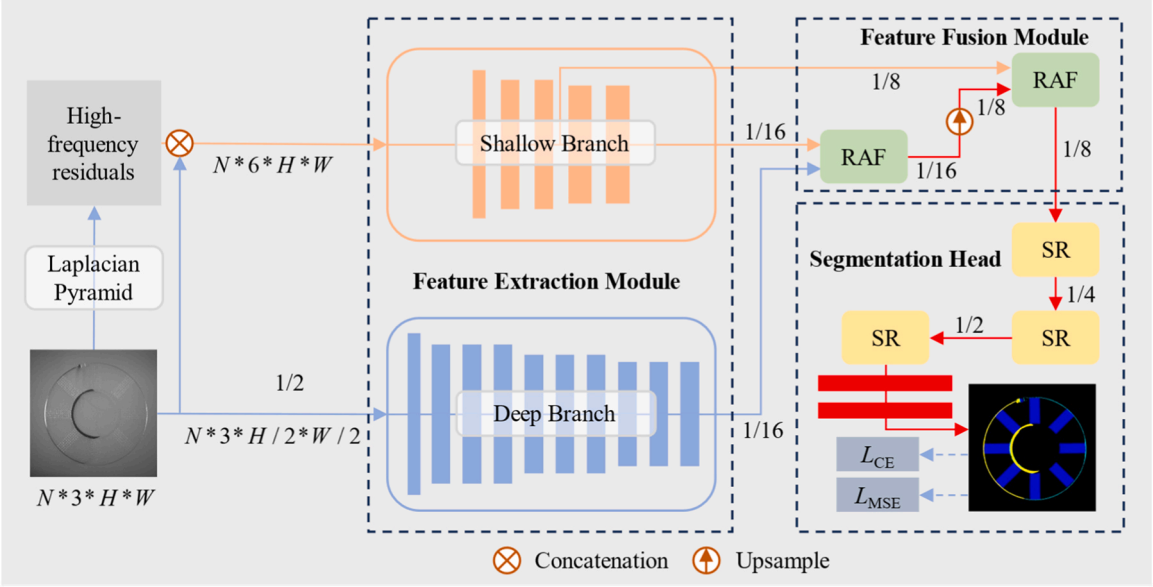

Fig. 2. Our improved ISDNet (including the designed SR module and the refined upsampling method) can be divided into three modules: feature extraction module, feature fusion module and segmentation head.

$$w^{\mathrm{FI}} = \mathrm{MLP}\big(\mathrm{Flatten}(\mathrm{R})\big) \tag{4}$$

Referring to residual connection, the channel attention weight is added to the feature interaction weight and modulated with a learnable parameter α. the sigmoid function is used to obtain the feature fusion weight as follow:

$$w_s = \mathrm{Sigmoid}\big(w_s^{\mathrm{CA}} + \alpha_s \times w_s^{\mathrm{FI}}\big) \tag{5}$$

$$w_d = \mathrm{Sigmoid}\big(w_d^{\mathrm{CA}} + \alpha_d \times w_d^{\mathrm{FI}}\big) \tag{6}$$

Finally, high-level features and low-level features are multiplied with their respective feature fusion weights, and the results are summed to obtain the final fused feature as follow:

$$F_{\mathrm{fusion}} = w_s \times F_s + \mathrm{Upsample}\big(w_d \times F_d\big), \tag{7}$$

where Upsample$(\cdot)$means upsampling operation to increase the feature map resolution. The upsampling operation is achieved through bilinear interpolation, which is a method for estimating the value of a function at any point inside a rectangle, given the values of the function at the four corners of the rectangle. It works by first performing linear interpolation in one direction, and then again in the other direction. The result is a weighted average of the four corner values, where the weights depend on the distance between the point and the corners.

During the training process, the standard cross-entropy loss is used to train the model, donated asL_{SEG}. Cross-entropy loss accepts the label and prediction probability as input to measures the difference between the true distribution and the predicted distribution, thus reflecting the accuracy of the model. This loss has been widely used in classification. Through minimizing its value, the model can be well trained. An auxiliary segmentation head is introduced in the deep branch, and it utilizes the feature map output by deep branch to generate the segmentation mask in advance and calculates cross-entropy loss (L_{AUX}) as part of the total loss. This design performs an additional supervision to the feature extraction process of deep branch, which motivates the deep branch to extract features efficiently to minimize the loss of downstream tasks. Besides, super resolution technique is applied to help reconstruct the original image and the super resolution loss (L_{SR}) and the structure distillation loss (L_{SD}) are special designed. The pipeline of ISDNet [31] is shown in Fig. 1.

The designs mentioned above enable ISDNet [31] to achieve a desirable trade-off between speed and accuracy when solving high-resolution image segmentation. However, there still exist some problems. After combining the deep branch and the shallow branch with the relation-aware feature fusion module, ISDNet [31] adopts a convolutional layer with 64 kernels to adjust the feature map channels, then applies a convolutional layer with the same number kernels with the categories to make every category' prediction, and then bilinearly upsamples the feature maps by eight times reverting to the original resolution. Such an operation is harmful to the detection of powder spreading defects. Narrow streak-like defects and small dot-like defects often appear on the powder bed and occupy only a very small area in the high-resolution powder bed images. This way of determining the category of pixels in a region by only a few pixels can easily lead to them being ignored. In addition, the upsampling amplitude is so large that the shape of the detected defect region will be relatively regular. However, powder spreading defects usually present a variety of complex shapes. This leads to unavoidable deviation in the defect region obtained by sampling on a large amplitude, which impairs the defect detection performance. Though the super resolution technique is applied in ISDNet [31] to help reconstruct the original image, but it works before the final upsampling process instead of right in the final upsampling process, which doesn't directly have an intuitive effect on the output. Due to the large upsampling amplitude of ISDNet [31], it is not capable of obtaining fine-grained powder spreading defect detection results and cannot be directly applied to the monitoring of powder spreading in SLM.

2.2. Model uncertainty estimation

Uncertainty in deep learning can be divided into two categories: aleatoric uncertainty and epistemic uncertainty. The former refers to the introduction of random noise due to the influence of external factors on data collection, labeling, etc. The latter refers to whether the model can correctly recognize the data [33]. In this paper, we mainly focus on epistemic uncertainty, which is also referred to model uncertainty. In original neural networks, the network weights are all fixed-point estimates, and certain inputs will only yield certain outputs. This certainty

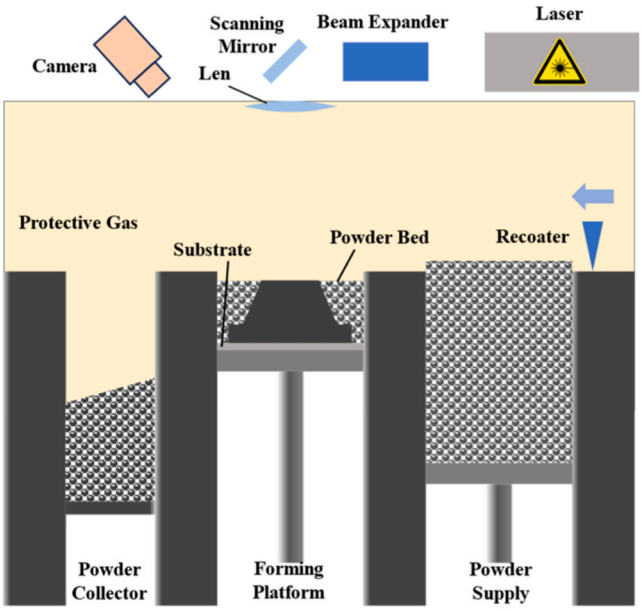

Fig. 3. Powder bed image acquisition.

mechanism does not allow for model uncertainty estimation. Bayesian inference sets the network weights $\theta = (W_1, \ldots, W_n)$ as probability distributions and applies Bayesian theory to derive the probability distribution of the output to achieve model uncertainty estimation. Specifically, for a given training dataset (x, y), assuming a priori distribution $p(\theta)$, the posterior distribution can be modeled as:

$$p(\theta|x, y) = \frac{p(y|x, \theta)p(\theta)}{p(y|x)} \propto p(y|x, \theta)p(\theta), \tag{8}$$

where $p(y|x)$ is called model evidence and is defined as:

$$p(y|x) = \int p(y|x, \theta)p(\theta)d\theta. \tag{9}$$

Once the posterior distribution on the network weights has been established, given a new input $x*$, the predictive distribution of the output $y*$ can be obtained as:

$$p(y* \mid x*, x, y) = \int p(y* \mid x*, \theta)p(\theta|x, y)d\theta. \tag{10}$$

The posterior probability is difficult to find because the network weights cannot be exhausted in practice. Therefore, variational inference and sampling methods are commonly used to approximate the posterior distribution. Variational inference [34] uses KL divergence to constrain the probability distribution $q(\theta)$ to approximate the posterior distribution. Monte Carlo-dropout [35] adds a dropout layer after each network weight to achieve random sampling, which has been proved to approximate the posterior distribution. But it decreases the capacity of the model and thus sometimes hurts performance.

In addition, model ensemble [36] is also a common method for model uncertainty estimation. Compared to Bayesian inference, model ensemble is easy to implement and has good predictive robustness. For a multi-classification semantic segmentation problem, suppose that there are C categories. Then a model will make C predictions for each pixel, corresponding C categories. After the model output is processed by the Softmax function, the prediction probability of each category will be obtained. And the index of the maximum probability is regarded as the true category of the pixel. For example, for (0.1, 0.2, 0.7), 0.7 is the maximum value, and its index is 2 (numbering starts from 0). For model ensemble, it combines multiple trained models to obtain multiple outputs with the same inputs. Suppose that there are M models used for ensemble and all outputs are processed by Softmax function. Then each category of a pixel will obtain M prediction probabilities. Their mean is regarded as the final prediction probability of the category. With the final prediction probabilities of C categories , the true category of a pixel is the index with the maximum probability and the prediction uncertainty in this pixel can be measured through the variance. This process is described as:

$$y = \frac{1}{M}\sum_{i=1}^{M} \mathrm{Softmax}\left(f_i(x)\right), \tag{11}$$

$$\hat{y} = \mathrm{Argmax}(y), \tag{12}$$

$$\mathrm{Var} = \mathbb{E}\left(y^2\right) - [\mathbb{E}(y)]^2. \tag{13}$$

where M donates the number of ensembled members, $f(\cdot)$ donates models used for ensemble, $\mathrm{Argmax}(\cdot)$ donates the operation to search the category index with the maximum prediction probability, y donates the final prediction probability, $\hat{y}$ donates the final prediction, $\mathbb{E}(\cdot)$ donates mathematical expectation, and Var donates variance.

How to obtain multiple trained models has been extensively studied, including random initialization of network weights, data augmentation, and network structure differences. However, it is time-consuming and laborious to train and tune multiple models. And the burden of memory and computation increases with the number of ensembled models, especially for powder bed images with high resolution inherently cause pressure on memory and computation. More importantly, because of random training, each model has different representation capacity, leading to the bias in the perception of the same input. A model may make predictions with low prediction accuracy as well as low model uncertainty, or high prediction accuracy as well as high model uncertainty. This is due to the mismatch between the prediction confidence and the prediction accuracy. Using a model with such bias for ensemble will influence the effectiveness of the model uncertainty estimation and make it difficult to exploit the uncertainty effectively later.

3. The proposed method

3.1. Improved ISDNet for powder spreading defect detection

As mentioned above, ISDNet demonstrates its advantages in high-resolution image segmentation, but its large upsampling amplitude results in the failure to obtain fine-grained defect detection results, limiting its application in powder-laying defect detection. Based on ISDNet, we propose a step-wise upsampling strategy combined with the super resolution technique to replace the original upsampling method for fine-grained identification.

Specifically, we introduce the Super Resolution (SR) module to ISDNet, for dividing the upsampling step to recover the original resolution into three same operations to complete. In the SR module, the input feature map is first upsampled twice in the nearest way. And then a convolutional layer with a kernel size of 3×3 and a stride of 1 is applied, followed by batch normalization and ReLU activation function. After that, a dropout function is added with a dropout rate of 0.2 to avoid overfitting. Last, a convolutional layer with a kernel size of 3×3 and a stride of 1 is applied. After the fusion of high-level features and low-level features, the feature map will be reverted to the original resolution through repeating the SR module three times. SR module limits each upsampling to twice, and then uses the convolution operation to aggregate the detailed information of the upsampled feature map, which supplements the information in the upsampling process and the connection between pixels is relatively continuous. Eight times upsampling is realized through three super-resolution modules, which is conducive to the detection of small powder coating defects. The range of detected defects is relatively continuous, which is conducive to the

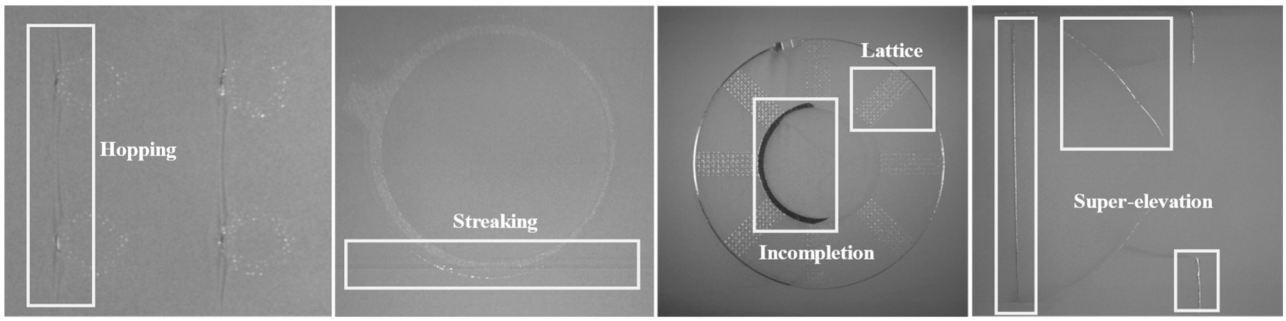

Fig. 4. Powder spreading defects: super-elevation, incompletion, hopping, streaking and lattice.

generation of fine-grained masks.

We give out the auxiliary segmentation head introduced into the deep branch as we find it not helpful for existing structures. Besides, we remain the standard cross-entropy loss (L_{CE}) for final segmentation results and add the mean- squared-error loss (L_{MSE}) for the SR module, further abandoning other setups in original ISDNet. The feature map outputted by the deep branch is upsambled by two times before inputting the relation-aware feature fusion module in ISDNet. But this operation is not combined with any other features. Thus, we replace it by downsampling the original image by two times instead of four times before inputting into the deep branch. Our improved ISDNet is shown in Fig. 2.

3.2. Uncertainty estimation

We utilize checkpoints to achieve model ensemble for model uncertainty estimation, which only requires a single training process. To reduce computational load raised by model ensemble, we adjust the network structure to achieve feature reuse. To eliminate cognitive bias in the ensembled checkpoints, we apply temperature scaling to calibration each checkpoint used for ensemble, leading to better uncertainty estimation.

3.2.1. Model ensemble

All parameters of a network saved during training process are called checkpoint or snapshot. It is often saved after several training epochs. Since the order of samples is random and parameters is updated in each epoch, each checkpoint contains inconsistent weights, which can be exploited for model ensemble. Adopting checkpoints to achieve model ensemble has been researched before. Huang et al. [37] exploited SGD to converge to and escape from local minima as the learning rate is lowered, which allow the model to visit several weight assignments to obtain checkpoints for ensemble. Chen et al. [38] explored the performance of checkpoint ensemble in a vanilla neural network, a convolutional neural network, and a long short term memory network. Garipov et al. [39] showed that the optima of deep neural networks are connected by simple pathways and proposed a training procedure to find these pathways for fast ensemble. Das et al. [40] presented the approach of using checkpoints to create ensembles of BERT-based transformers, which improved the performance of classification. Wang et al. [41] proposed a novel method to ensemble the checkpoints, where a boosting scheme is utilized to accelerate model convergence and maximize the checkpoint diversity. Due to the convenience of not having to train multiple times, we adopt checkpoints to achieve model ensemble for model uncertainty estimation. On the premise that the training process can converge, a relatively large learning rate is set so that the network can achieve better performance at multiple stages of the training process, and multiple sets of checkpoints are obtained for model ensemble. This mode only requires training and tuning of a single model, which greatly reduces the workload. The overall procedure is summarized in Algorithm 1.

To alleviate the problem of slow inference caused by multiple forward propagations, we restructure the individual models and reduce the computational time via feature reuse. The network structure of the modified ISDNet mentioned above can be divided into three parts: the feature extraction module, the feature fusion module, and the segmentation head. Among them, the feature extraction module uses two network branches, resulting in a large amount of computational time. In comparison, the network structure of the feature fusion module and the segmentation head is quite lightweight. Therefore, we adapt the model structure to a feature extraction module, a feature fusion module, and multiple segmentation heads to achieve model ensemble. The forward propagation of model ensemble can be formulated as:

$$y = \frac{1}{M}\sum_{i=1}^{M} \mathrm{Softmax}(S_i(F_1(H_1(x)))), \tag{14}$$

where $H(\cdot)$ is the feature extraction module, $F(\cdot)$ is the feature extraction module, $S(\cdot)$ is the segmentation head,M donates the number of ensembled members, and the best checkpoint number by default is 1. This approach to achieve model ensemble using only lightweight structures can significantly reduce the computational complexity compared to using the full network. To ensure effective feature extraction and fusion, the best performing checkpoint is applied in the feature extraction module and the feature fusion module. As for the segmentation heads, all checkpoints are sorted in descending order according to their performance on the evaluation metrics, and the top M checkpoints are chosen according to the scale of model ensemble. The ensemble output $y$is obtained according to Eq. 9. Model uncertainty is then calculated according to Eq. 10.

3.2.2. Model calibration

In addition, because of random training, each model has different representation capacity, leading to the bias in the perception of the same input. That is, the trained model would be overconfident or diffident. The biased model will influence the estimation of uncertainty. Thus, temperature scaling is used to calibrate each model used for ensemble to better uncertainty estimation. For a binary classifier, a well-calibrated model should be that its predicted probability is consistent with the actual empirical probability [42], which can be defined as:

$$\mathbb{P}(\widehat{Y} = Y|\widehat{P} = p) = p, \forall p \in [0,1] \tag{15}$$

where $\widehat{P}$denotes the prediction confidence, $\widehat{Y}$denotes the prediction category, and Ydenotes the label. An example of the well-calibration model is that a model predicts a probability of 0.6 that it will rain in the next 10 days, and then it does rain on 6 days of those 10 days.

To judge how well a classification model is calibrated, reliability diagram and expected calibration error (ECE) are widely adopted [43]. Divide the confidence interval of 0–1 into K bins and assume that B_k donates the bin of samples with prediction confidence in the range of $\left(\frac{k-1}{K}, \frac{k}{K}\right]$. The average confidence on bin B_k is calculated as:

$$\mathrm{Conf}_k = \frac{1}{|B_k|}\sum_{i\in B_k} p(\hat{y}_i = y_i|x_i,\theta), \tag{16}$$

whereθdonates the model parameters and $p(\hat{y}_i = y_i|x_i,\theta)$ means prediction confidence made by each sample in B_k through model parametersθ. The expected accuracy on each bin B_k is calculated as:

$$\mathrm{Acc}_k = \frac{1}{|B_k|}\sum_{i\in B_k} \mathbb{I}(\hat{y}_i = y_i), \tag{17}$$

where$\mathbb{I}(\cdot)$is indicator function, and its value is one when the prediction is correct, otherwise it is zero. With the$M$pairs of values, reliability diagram can be plotted, in which confidence and accuracy are used as horizontal and vertical axes respectively. Besides, The ECE is calculated as:

$$\mathrm{ECE} = \sum_{k=1}^{M}\frac{|B_k|}{N}|\mathrm{Acc}_k - \mathrm{Conf}_k|, \tag{18}$$

where$N$donates the number of samples. The closer the curve of the reliability plot is to the diagonal and the smaller the value of ECE, the better the model is calibrated.

As for multiclass classifier, the category with the highest confidence is regarded as the final prediction and its confidence is regarded as the prediction confidence. Based on whether the prediction is correct or not, the problem is reduced to the binary classifier [44].

The checkpoint for model ensemble is calibrated here with temperature scaling [42] because it is easy to implement and works well. For a given dataset(X,Y), all samples are inputted to the improved ISDNet and their corresponding outputs are defined as follow:

$$Z = S(F(H(X))) \tag{19}$$

Then, a temperature factor $T$is introduced to the Softmax function and applied to$Z$for confidence outputs, described as:

$$Y' = \mathrm{Softmax}(Z/T) \tag{20}$$

After that, the negative log likelihood loss L_{NLL} between sample labels Y and confidence outputs $Y'$is minimized to get the proper temperature factor. Note that the temperature factor that minimizes the negative log-likelihood loss does not necessarily minimize the ECE. We only use the process of minimizing the negative log likelihood to find the appropriate temperature factor. After calibration, the output of the proposed model ensemble is described as:

$$y = \frac{1}{M}\sum_{i=1}^{M}\mathrm{Softmax}(S_i(F_1(H_1(y))/T_i), \tag{21}$$

where the best checkpoint number by default is 1. The overall procedure is summarized in Algorithm 1.

Algorithm 1. Model Uncertainty Estimation with Model Ensemble and Model Calibration.

1: Train and tune the improved ISDNet.
▷ Calculate the evaluation metrics score on the validation set and save the checkpoint after every epoch training.
2: Select the best M checkpoints for model ensemble and arrange them in descending order.
▷ Donated as $\{ck_1, ck_2, \ldots, ck_M\}$.
3: Apply temperature scaling to the M checkpoints on the validation set.
for $i = 1 : M$ **do**
$Z \leftarrow S_i(F_i(H_i(X)))$
$Y' = \mathrm{Softmax}(Z/T)$
Minimize Negative Log Likelihood Loss to get the proper temperature factor.
▷ $T_i \leftarrow \min L_{\mathrm{NLL}}(Y, Y')$
end for
4: Adjust the improved ISDNet structure to achieve model ensemble.
▷ $f(x) = S(F(H(x))) \rightarrow y = \frac{1}{M}\sum_{i=1}^{M}\mathrm{Softmax}(S_i(F_1(H_1(x)))/T_i)$
5: Obtain the final prediction and model uncertainty.
▷ The prediction is calculated through apply argmax to y along the channel dimension.
$y_{\mathrm{pred}} = \mathrm{Argmax}(y)$
▷ the model uncertainty is estimated by the variance of the ensemble output y.
$\mathrm{Var} = \mathrm{E}(y^2) - [\mathrm{E}(y)]^2$

3.3. Uncertainty-driven model improvement

Model uncertainty indicates the extent to which the model is cognizant of the inputs, providing additional information for decision making, and thus beyond only uncertainty estimation, we make a further step towards improving the model performance via uncertainty information to achieve a more trustworthy defect detection. We observe two types of regions with high model uncertainty on the segmentation mask: regions where the model produces misclassification and the contour edges of objects. The former demonstrates that the model is incapable of cognizing the region correctly and requires the intervention of a decision expert when it reaches a certain level. The latter is because the contour edge is in the transition region between the two textures, which is naturally difficult to distinguish and inevitably generates a high level of uncertainty. We also observe that the background is the best predicted class, and the true label of the misclassified region tends to be consistent with the class of the surrounding region in exception to the

background. Therefore, we propose an uncertainty-driven model improvement scheme that utilizes model uncertainty for improving segmentation performance as follow:

Algorithm 2. Uncertainty-driven Model Improvement.

1: Input the powder bed image to the improved ISDNet to obtain its prediction y_{pred} and model uncertainty heatmap u respectively.
2: Normalize the model uncertainty heat map and set proper threshold to extract the region with high uncertainty.
▷ $\overline{u}_{i,j} = \dfrac{u_{i,j} - u_{\min}}{u_{\max} - u_{\min}},\ 1 \le i \le H,\ 1 \le j \le W$.
▷ $\delta = \dfrac{\overline{u}_{1,1} + \overline{u}_{1,W} + \overline{u}_{H,1} + \overline{u}_{H,W}}{4} + 0.05$
▷ The region with high uncertainty: $u_{\text{HU}} = \{\overline{u} > \delta\}$.
3: Apply connected domain detection to the region u_{HU} .
▷ Divide the region u_{HU} into smaller ones: $\{u^1_{\text{HU}}, u^2_{\text{HU}}, \ldots, u^k_{\text{HU}}\} \leftarrow u_{\text{HU}}$.
▷ Obtain the circumscribing rectangle for each small region: $b_k \leftarrow u^k_{\text{HU}}$.
4: Count the pixel number of each category in the rectangle region b_k on the prediction y_{pred} and the category except background with the most pixels is regarded as the true category of the region u^k_{HU} .
▷ $C_k \leftarrow \underset{C \neq 0}{\text{Argmax}}(b_k \cap y_{\text{pred}})$, 0 donates background.
5: Modify the prediction on the prediction y_{pred} .
▷ If C_k is lattice, then $(u^k_{\text{HU}} \cap y_{\text{pred}}) \leftarrow C_k$

First, the uncertainty heatmap obtained by model ensemble is normalized and a proper threshold δ, is set to extract regions with high uncertainty as follow:

$$\overline{u}_{i,j} = \frac{u_{i,j} - u_{\min}}{u_{\max} - u_{\min}}, 1 \le i \le H, 1 \le j \le W \tag{22}$$

$$\delta = \frac{\overline{u}_{1,1} + \overline{u}_{1,W} + \overline{u}_{H,1} + \overline{u}_{H,W}}{4} + 0.05 \tag{23}$$

where H and W represent the height and weight of the heatmap, respectively, u_{ij} donates the pixel located in the position $(i.j)$ of the uncertainty heatmap, $u_{\min}$ and $u_{\max}$ represent the minimum value and maximum value of the uncertainty heatmap, and $\overline{u}_{ij}$ donates the pixel located in the position $(i.j)$ of the normalized uncertainty heatmap. For determining the threshold of the high uncertainty region, we refer to the uncertainty value of the background region for determination. The threshold is set by the values of the four corners of each uncertainty heatmap, which are the most accessible background regions. Here the high uncertainty region u_{HU} is obtained.

Then we apply connected domain detection to these regions with high uncertainty to further refine the high uncertainty regions and avoid mistakes in the judgement of the true category if the region is too large. Then, the circumscribing rectangle b_k for each connected domain u^k_{HU} is obtained, and the pixels of each category are counted inside the rectangular box on the mask. The category with the highest number of pixels, that is not the background, is treated as the true category C_k of that high uncertainty region, which is modified on the predicted result. The overall procedure is summarized in Algorithm 2.

4. Experiments and discussion

4.1. Powder spreading defect dataset

For powder bed image acquisition, the industrial camera is arranged off-axis on top of the chamber to capture images from the side, shown in Fig. 3. The acquired images are adjusted to the front view by perspective transformation, and then cropped to remove excess region and keep only the powder bed region. We collected some powder bed images of SLM from our collaborator (Xi'an Kongtian Electromechanical Intelligent Manufacturing Co., Ltd.) and labeled them at the pixel level. Due to the small layer thickness of the powder spreading, there is little difference between the powder bed layers, leading to many similar images. Furthermore, the occurrence frequency of some defects is low so that the collected images is limited. The dataset ended up with 406 images. The resolutions of these images span from 1400 * 1400–4300 * 4300. For the training convenience, we downsample it to 1024 * 1024. Totally, five types of defects were included in the dataset, which were named: super-elevation, incompletion, hopping, streaking, and lattice, and shown in Fig. 4.

The occurrence of super-elevation is because the molten layer is too high for the metal powder to completely cover it, which is caused by the thermal stress, and often occurs with the edge of the contour. Incompletion is due to insufficient metal powder amount or abnormality of the powder spreading device, which results in forming areas not covered by metal powder. Hopping is caused by the collision between the recoater and the protrusion on the powder bed, which is featured by the stripes on the powder bed perpendicular to the direction of recoater movement. Streaking is resulted from recoater damage or large contaminants on the

Table 1
Number of images for each category.

Categories	Background	Super-elevation	Incompletion	Hopping	Streaking	Lattice
Number	406	377	52	193	229	134

Table 2
Defect segmentation performance comparison between ISDNet and our method on the testing set.

Categories		Background	Super-elevation	Incompletion	Hopping	Streaking	Lattice
ISDNet	CPA	0.9976	0.6097	0.9673	0.5273	**0.5179**	**0.9158**
Ours		**0.9980**	**0.7918**	**0.9760**	**0.5385**	0.4902	0.8482
ISDNet	IoU	**0.9928**	0.4734	0.8606	0.4211	**0.4427**	**0.8639**
Ours		0.9927	**0.6298**	**0.8819**	**0.4657**	0.4289	0.8054
ISDNet	MPA	0.7559					
Ours		**0.7738**					
ISDNet	MIoU	0.6757					
Ours		**0.7008**					
ISDNet	Time (per 46 images)	5 s					
Ours		6 s					

powder bed dragged by recoater, which is characterized by the stripes on the powder bed parallel to the direction of recoater movement. Lattice is a special shape left on the powder bed by the support set to ensure the forming quality, and it is not a defect in the strict sense. However, to effectively distinguish other defects, it is also classified as a special defect. The common point-support and sheet-support are vaguely visible in full on the powder bed, often in large area. Since there are multiple defects on one powder bed image, the sum of the number of images for each defect is greater than the number of images in the dataset. The number of images for each category in the dataset is shown in Table 1.

4.2. Evaluation metrics and detailed settings

We randomly divide the powder spreading defect dataset into three parts for training, validation, and testing by 8:1:1. Due to the small dataset scale, we make a manual adjustment to ensure that each type of defect has a certain number in each set to avoid that a set does not contain certain defects. Finally, 320 images are used for training, 40 images for validation, and 46 images for testing. For ISDNet, we used its official code, and the hyperparameter settings are consistent with those in the corresponding paper. Besides, we have implemented our method using PyTorch with the changes mentioned above. The learning rate is set to 0.01 and held constant. The optimizer selects Adam, the model is trained for 1000 rounds, and the network weights are saved for each round of training. All results are obtained by running on an RTX 3090 GPU.

For evaluation metrics, in addition to calculating Class Pixel Accuracy (CPA) and Intersection over Union (IoU) for each category, we also used Mean Pixel Accuracy (MPA) and Mean Intersection over Union (MIoU) to evaluate the defect classification and location performance. The definitions of evaluation metrics are as follows:

$$\mathrm{CPA} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{24}$$

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{25}$$

$$\mathrm{MPA} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{CPA}_i, \tag{26}$$

$$\mathrm{MIoU} = \frac{1}{C}\sum_{i=1}^{C} \mathrm{IoU}_i, \tag{27}$$

where TP means True Positive, FP means False Positive, FN means False Negative, and C donates the number of categories.

4.3. Defect segmentation result

Our model eventually achieves MPA of 0.8169 and MIoU of 0.7551 on the validation set, while ISDNet achieves MPA of 0.7832 and MIoU of 0.7011. The performance of the two methods on the test set is shown in Table 2, demonstrating that our improvements can significantly improve the segmentation performance. Among them, the detection performance of super-elevation, incompletion and hopping has been significantly improved. The improvement of the first two is very important for powder spreading monitoring. Super-elevation will accumulate during the forming process, which may lead to the formation of hopping on the one hand, and on the other hand, such defect may even lead to processing failure when they reach a certain area. If incompletion is not detected in time and occurs in several consecutive layers, it will affect the forming accuracy and lead to scrap of the formed part. Compared to ISDNet, although our method has three extra super resolution modules, the impact on the inference speed is negligible.

The improvement in super-elevation segmentation performance is attributed to the refined upsampling process, where the super-resolution module reorganizes the features, resulting in a finer-grained mask compared to the direct eight-fold upsampling practice of ISDNet. This is reflected in the more continuous detection of defect areas and the effective detection of defects with small areas, as shown in Fig. 5. It is also the reason for the degraded performance of lattice segmentation. The lattice segmented by ISDNet presents the characteristic of a complete area, while our method leads to the appearance of holes. However, for some dense dotted super-elevation, ISDNet is preferred to identify them as lattice, and our method can better discriminate them. Incompletion has the feature of a relatively large defective area, which is easy to confuse with super-elevation when the area of super-elevation is large. Our approach has improved the performance for the segmentation of super-elevation and indirectly helped the segmentation of incompletion. The reason for the poor performance of both methods for the segmentation of streaking is related to the morphology of the streaking itself. Streaking often appears as a shallow stripe on high-resolution powder bed images, making it difficult to confirm its edges when labeling. Streaking is sometimes dense and difficult to identify completely, so we can only label the obvious stripes. By comparing the mask with the original image, we find that the model still performs quite well in recognizing stripes. Even unlabeled stripes can be recognized, which leads to a poor performance on the metrics.

4.4. Uncertainty estimation

During the training process of the improved ISDNet, the network weights of each training round are saved as checkpoint. As for the checkpoint selection, we define:

$$\mathrm{Score} = 0.5 \times \mathrm{MPA} + 0.5 \times \mathrm{MIoU}. \tag{28}$$

The scores of all checkpoints are calculated and the results are sorted in descending order. Too large a scale of model ensemble will lead to an increase in computation burden, and too small a scale will not manifest the diversity of predictions. Hence, we set the scale of model ensemble to 5. The highest scoring checkpoint is applied to the feature extraction and fusion modules, and the top five scoring checkpoints are applied to each of five segmentation heads.

The selected checkpoint is calibrated using temperature scaling prior to ensemble, as shown in Algorithm 1. Temperature scaling is performed

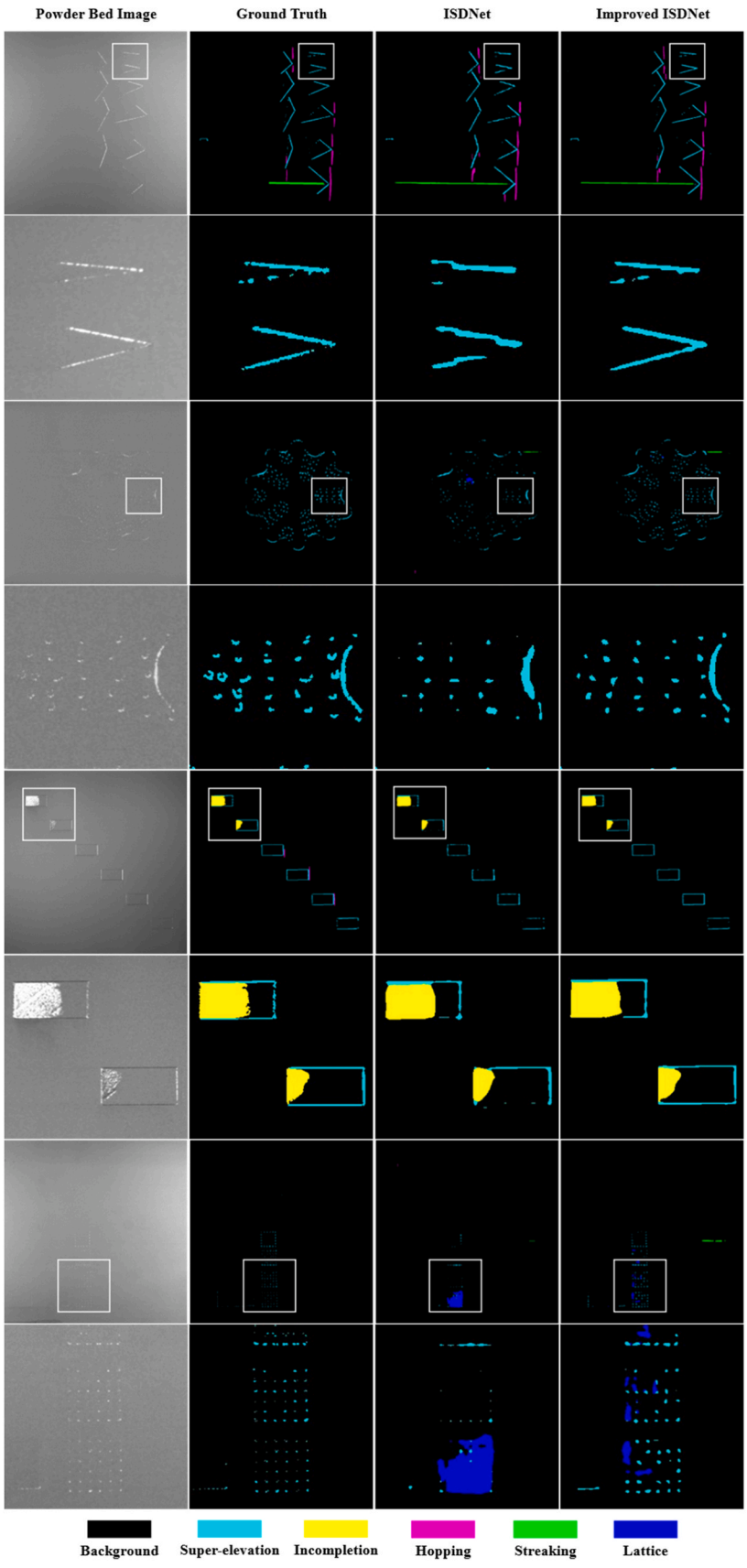

Fig. 5. Segmentation visualization comparison for both our method and ISDNet.

Table 3
Performance of selected checkpoints and the corresponding temperature factors.

Checkpoints	MPA	MIoU	Score	Temperature	ECE Before Calibration	ECE After Calibration
#1	0.8169	0.7551	0.7860	0.8324	0.00146116	0.00064222
#2	0.8140	0.7562	0.7851	0.7629	0.00144244	0.00083944
#3	0.8155	0.7510	0.7835	0.8526	0.00125335	0.00060728
#4	0.8124	0.7532	0.7828	0.5887	0.00136726	0.00035785
#5	0.8098	0.7540	0.7819	0.8567	0.00146539	0.00085154

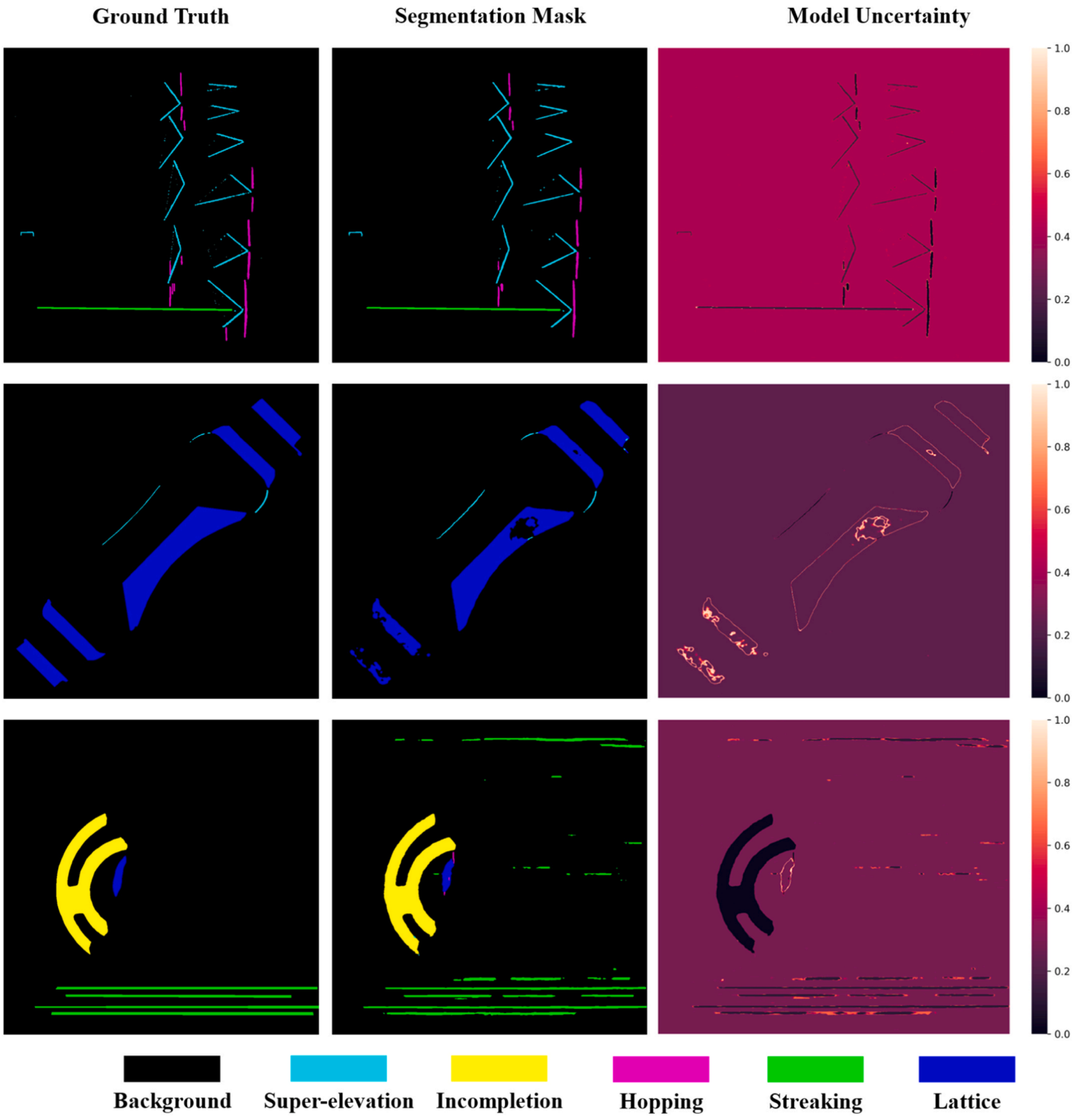

Fig. 6. Model Uncertainty Heatmap.

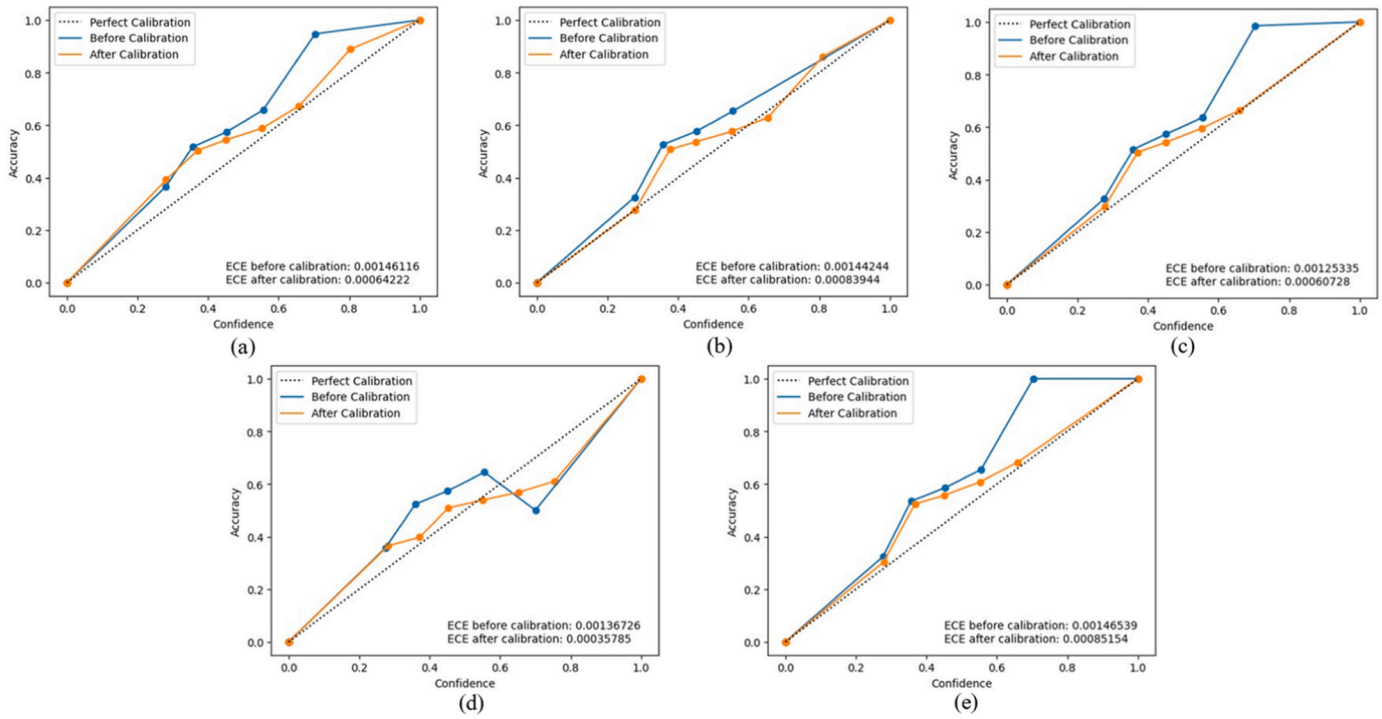

Fig. 7. Comparison of reliability diagram before and after calibration.

on the validation set, a temperature factor is introduced in the softmax function for smoothing the prediction confidence, the negative log-likelihood loss is used as the optimization objective, and Adam is used for the optimizer. The performance and temperature factors of the checkpoints we selected for integration are shown in the table below. The temperature factors in Table 3 are all less than one, indicating that the model is biased with underconfidence. The prediction confidence of the model can be appropriately increased by introducing a temperature factor less than one. We divided the confidence interval from zero to one into ten bins and plotted the reliability diagram before and after calibration for each checkpoint separately, as shown in Fig. 7. The fold line before calibration was located above the diagonal, which means that the model does not behave confidently enough. After temperature scaling, the fold of the reliability plot is closer to the diagonal and the ECE is significantly reduced. That is the model is calibrated somewhat. Besides, we use a third-party library for PyTorch, called *thop*, to calculate the number of floating point operations (FLOPs) when the improved ISDNet is inputted a powder bed image with a resolution of 1024 * 1024. Our improved ISDNet performs almost 165.1 GFLOPs while model ensemble performs almost 299.4 GFLOPs with 5 checkpoints. In the normal way of model ensemble, the FLOPs would be 5 times more than that of the original model. However, via adjusting the network structure to achieve feature reuse, the FLOPs are less than twice as much as the original model.

Utilizing the checkpoints selected above and the corresponding temperature factors, we implement model ensemble and measure the model uncertainty more precisely, and the results are shown in Fig. 6. It can be clearly seen that there is higher uncertainty at the edges of the contours of the objects. In addition, there is also a high uncertainty in the region of the model misclassification, which does not completely cover the misclassified region, but encompasses the edges of the misclassified region. This information can be leveraged to improve the segmentation performance of defects and inform subsequent decisions. Furthermore, the model has low uncertainty for super-elevation and incompletion, indicating that the model perceives both well, which is related to the fact that these two defects present a metallic luster and are better distinguished from the background. Defects like streaking, hopping and lattice are more like the background and easy to misjudge. Among them, lattice will show a larger area of misclassification and contain more high uncertainty areas. And as the reason mentioned before, streaking's edge region presents high uncertainty. Besides, we compare the defect segmentation performance between improved ISDNet and model ensemble, and the results are shown in Table 4. Although model ensemble will cause a little performance loss, it can easily achieve model uncertainty estimation and help evaluate the credibility of predictions.

For whether the checkpoint needs to be calibrated for better uncertainty estimation, we made separate uncertainty heatmaps before and after calibration, as shown in Fig. 8. It can be clearly seen that the ensemble using the calibrated checkpoints can make the uncertainty levels show a more obvious stratification effect, which is reflected in the fact that the background areas can be more clearly distinguished from the high uncertainty areas, facilitating the extraction of high uncertainty areas.

4.5. Uncertainty-driven improvement

Regions with high model uncertainty are often found at the contour edges of objects and in misclassified regions, most prominently within the regions of lattice. Therefore, the post-processing method based on uncertainty here is for lattice area. The powder bed image is inputted into the model to get the prediction mask and uncertainty heatmap. The uncertainty heatmap is processed as shown in Algorithm 2.

The threshold is set by the values of the four corners of each uncertainty heatmap, which are the most accessible background regions. The comparison of the masks before and after post-processing is shown in Fig. 9. The proposed uncertainty-driven improvement method can improve the segmentation performance of lattice to some extent. However, this enhancement is relatively limited because the misclassified regions are not completely covered, and only the edge regions of the misclassified regions are encompassed. Our work shows that exploiting uncertainty for defect segmentation is helpful to improve model performance. It is worthwhile to continue exploring how to improve defect segmentation based on model uncertainty.

Table 4
Defect segmentation performance comparison between improved ISDNet and model ensemble.

Categories		Background	Super-elevation	Incompletion	Hopping	Streaking	Lattice
Improved ISDNet	CPA	0.9980	**0.7918**	**0.9760**	**0.5385**	**0.4902**	**0.8482**
Ensemble		**0.9983**	0.7547	0.9631	0.5325	0.4799	0.8397
Improved ISDNet	IoU	0.9927	0.6298	0.8819	0.4657	0.4289	0.8054
Ensemble		0.9927	**0.6299**	**0.8834**	0.4631	0.4238	0.8019
Improved ISDNet	MPA	**0.7738**					
Ensemble		0.7614					
Improved ISDNet	MIoU	**0.7008**					
Ensemble		0.6991					
Improved ISDNet	Time (per 46 images)	6 s					
Ensemble		7 s					

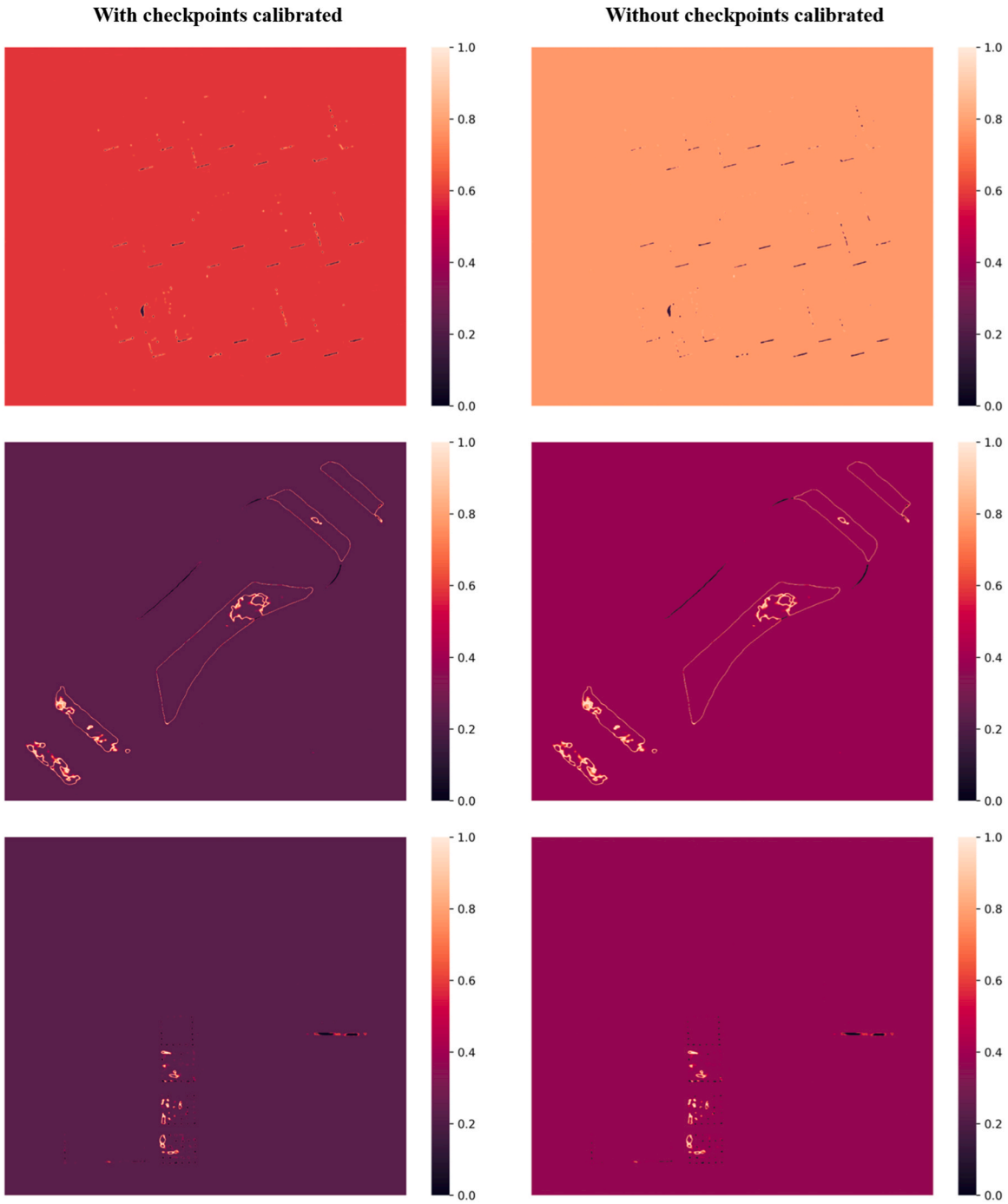

Fig. 8. Comparison of whether checkpoints are calibrated or not for uncertainty estimation.

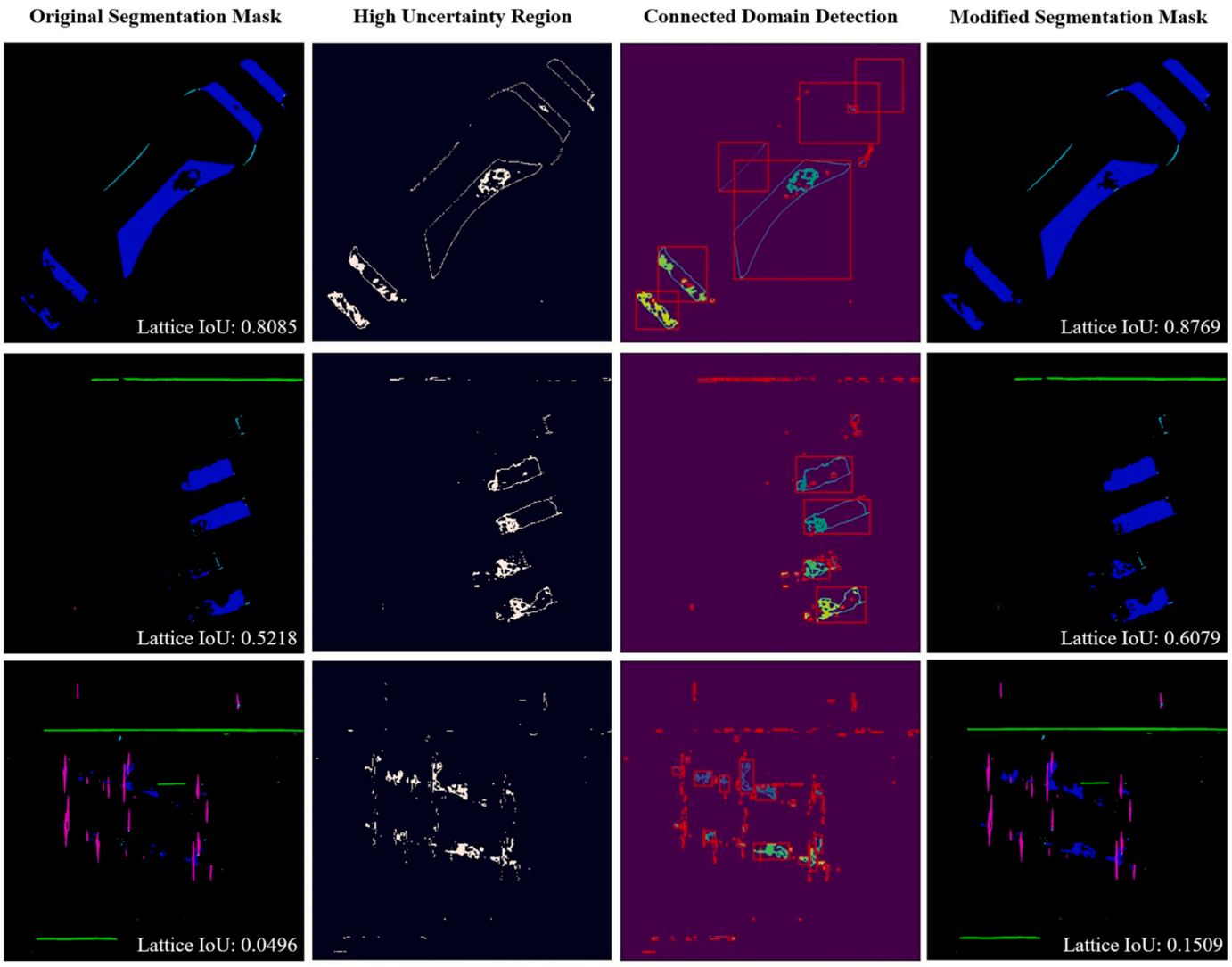

Fig. 9. Lattice segmentation improvement with model uncertainty.

5. Conclusions

To address the problem of unreliability in monitoring the quality powder spreading of SLM, we improve ISDNet for defect segmentation of high-resolution powder bed images, and the proposed super resolution module can effectively improve the segmentation performance of the model and obtain fine-grained masks by refining the upsampling process. Checkpoint ensemble is adopted for better achieving model uncertainty estimation, using checkpoints saved in a single train process to quickly achieve ensemble, utilizing feature reuse to reduce computational load carried by ensemble, and eliminating cognitive bias for each ensemble member by calibration with temperature scaling. The uncertainty region is effectively distinguished and facilitates post-processing after model calibration. An uncertainty-driven model improvement approach to improve the performance of partial defect segmentation based on uncertainty is also proposed. Our work achieves an effective segmentation of powder spreading defects and evaluates the segmentation results, further applying uncertainty to improve the performance of defect segmentation. Uncertainty estimation gives a reference of whether the prediction is trustworthy and helps making safe decisions. In the future, we will utilize uncertainty estimation in this work to control the powder spreading in SLM.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Key R&D Program of China (No. 2022YFB4600800) and by the National Natural Science Foundeation of China (No. 52105116).

References

[1] He W, Shi W, Li J, Xie H. In-situ monitoring and deformation characterization by optical techniques; part I: laser-aided direct metal deposition for additive manufacturing. Opt Laser Eng 2019;122:74–88.
[2] Nagarajan B, Hu Z, Song X, Zhai W, Wei J. Development of micro selective laser melting: the state of the art and future perspectives. Engineering 2019;5:702–20.
[3] Gu Dongdong, Zhang Hongmei, Chen Hongyu, Zhang Han, Xi Lixia. Laser additive manufacturing of high-performance metallic aerospace components. Chin J Lasers 2020;47:0500002.
[4] Zhu JH, Zhou H, Wang C, Zhou L, Yuan S, Zhang W. Status and future of topology optimization for additive manufacturing. Aeronaut Manuf Technol 2020;63:24–38.
[5] Zhao D, Lin F. A review of on-line monitoring techniques in metal powder bed fusion processes. China Mech Eng 2018;29:2100.

[6] Mandloi K, Amrapurkar P, Cherukuri HP. Discrete element modeling of scraping process and quantification of powder bed quality for slm. American Society of Mechanical Engineers. 2020. pp. V001T01A037.
[7] Neef A, Seyda V, Herzog D, Emmelmann C, Sch M, Nleber O, Kogel-Hollacher M. Low coherence interferometry in selective laser melting. Phys Procedia 2014;56:82–9.
[8] Zhang B, Ziegert J, Farahi F, Davies A. In situ surface topography of laser powder bed fusion using fringe projection. Addit Manuf 2016;12:100–7.
[9] DePond PJ, Guss G, Ly S, Calta NP, Deane D, Khairallah S, Matthews MJ. In situ measurements of layer roughness during laser powder bed fusion additive manufacturing using low coherence scanning interferometry. Mater Des 2018;154:347–59.
[10] Cao M, He P, Li Z. Online measurement technology for flatness and profile of metal additive manufacturing process. Foundry Technol 2019;40:40–6.
[11] T. Craeghs, S. Clijsters, E. Yasa, J. Kruth, Online quality control of selective laser melting, University of Texas at Austin, 2011.
[12] Zur Jacobsm J, Hlen U, Kleszczynski S, Schneider D, Witt G. High resolution imaging for inspection of laser beam melting systems. IEEE; 2013. p. 707–12.
[13] Abdelrahman M, Reutzel EW, Nassar AR, Starr TL. Flaw detection in powder bed fusion using optical imaging. Addit Manuf 2017;15:1–11.
[14] Lin Z, Lai Y, Pan T, Zhang W, Zheng J, Ge X, Liu Y. A new method for automatic detection of defects in selective laser melting based on machine vision. Materials 2021;14.
[15] Scime L, Beuth J. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Addit Manuf 2018;19:114–26.
[16] Scime L, Beuth J. A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. Addit Manuf 2018;24:273–86.
[17] Scime L, Siddel D, Baird S, Paquit V. Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation. Addit Manuf 2020;36:101453.
[18] Chen H, Lin C, Horng M, Chang L, Hsu J, Chang T, Hung J, Lee R, Tsai M. Deep learning applied to defect detection in powder spreading process of magnetic material additive manufacturing. Materials 2022;15.
[19] Mehta M, Shao C. Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. J Manuf Syst 2022;64:197–210.
[20] Fischer FG, Zimmermann MG, Praetzsch N, Knaak C. Monitoring of the powder bed quality in metal additive manufacturing using deep transfer learning. Mater Des 2022;222:111029.
[21] W. Chen, Z. Jiang, Z. Wang, K. Cui, X. Qian, Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images, 2019, pp. 8924–8933.
[22] Shan L, Li M, Li X, Bai Y, Lv K, Luo B, Chen S, Wang W. UHRSNet: a semantic segmentation network specifically for ultra-high-resolution images. Los Alamitos, CA, USA: IEEE Computer Society; 2021. p. 1460–6.
[23] Q. Li, W. Yang, W. Liu, Y. Yu, S. He, From contexts to locality: Ultra-high resolution image segmentation via locality-aware contextual correlation, 2021, pp. 7252–7261.
[24] Wang H, Tao C, Qi J, Xiao R, Li H. Avoiding negative transfer for semantic segmentation of remote sensing images. IEEE T Geosci Remote 2022;60:1–15.
[25] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, Icnet for real-time semantic segmentation on high-resolution images, 2018, pp. 405–420.
[26] H.K. Cheng, J. Chung, Y. Tai, C. Tang, Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement, 2020, pp. 8890–8899.
[27] C. Huynh, A.T. Tran, K. Luu, M. Hoai, Progressive semantic segmentation, 2021, pp. 16755–16764.
[28] T. Shen, Y. Zhang, L. Qi, J. Kuen, X. Xie, J. Wu, Z. Lin, J. Jia, High quality segmentation for ultra high-resolution images, 2022, pp. 1310–1319.
[29] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: multi-path refinement networks for high-resolution semantic segmentation, 2017, pp. 1925–1934.
[30] Zhang Z, Lu M, Ji S, Yu H, Nie C. Rich CNN features for water-body segmentation from very high resolution aerial and satellite imagery. Remote Sens-Basel 2021;13:1912.
[31] S. Guo, L. Liu, Z. Gan, Y. Wang, W. Zhang, C. Wang, G. Jiang, W. Zhang, R. Yi, L. Ma, Others, Isdnet: integrating shallow and deep networks for efficient ultra-high resolution segmentation, 2022, pp. 4361–4370.
[32] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, X. Wei, Rethinking BiSeNet for real-time semantic segmentation, 2021, pp. 9716–9725.
[33] Kendall A, Gal Y. What uncertainties do we need in bayesian deep learning for computer vision? Adv Neural Inf Process Syst 2017;30.
[34] Graves A. Practical variational inference for neural networks. Adv Neural Inf Process Syst 2011;24.
[35] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: representing model uncertainty in deep learning. PMLR 2016;pp. 1050:1059.
[36] Rahaman Rahul, Thiery Alexandre H. Uncertainty quantification and deep ensembles. Adv Neural Inf Process Syst 2021;34:20063–75.
[37] Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ. Snapshot ensembles: train 1. Get M Free 2017.
[38] H. Chen, S. Lundberg, S. Lee, Checkpoint ensembles: ensemble methods from a single training process, arXiv preprint arXiv:1710.03282, (2017).
[39] T. Garipov, P. Izmailov, D. Podoprikhin, D.P. Vetrov, A.G. Wilson, Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs, 2018.
[40] K.A. Das, A. Baruah, F.A. Barbhuiya, K. Dey, KAFK at SemEval-2020 Task 12: Checkpoint Ensemble of Transformers For Hate Speech Classification, Barcelona (online), 2020, pp. 2023–2029.
[41] Wang Feng, Wei Guoyizhe, Liu Qiao, Ou Jinxiang, Wei Xian, Lv Hairong. Boost neural networks by checkpoints. Adv Neural Inf Process Syst 2021;34:19719–29.
[42] Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. PMLR 2017:1321–30.
[43] Mehrtash A, Wells WM, Tempany CM, Abolmaesumi P, Kapur T. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. IEEE T Med Imaging 2020;39:3868–78.
[44] C. Gupta, A. Ramdas, Top-label calibration and multiclass-to-binary reductions, 2022.