Journal of Materials Processing Tech. 304 (2022) 117541

Contents lists available at ScienceDirect

Journal of Materials Processing Tech.

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

A part-scale, feature-based surrogate model for residual stresses in the laser powder bed fusion process

Guoying Dong [a,b,*], Jian Cheng Wong [c], Lucas Lestandi [c], Jakub Mikula [c], Guglielmo Vastola [c], Mark Hyunpong Jhon [c], My Ha Dao [c], Umesh Kizhakkinan [b], Clive Stanley Ford [c], David William Rosen [b]

[a] Department of Mechanical Engineering, University of Colorado Denver, Denver, CO, USA
[b] Digital Manufacturing and Design Center, Singapore University of Technology and Design, Singapore, Singapore
[c] Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore, Singapore

ARTICLE INFO

Associate Editor:Dr Jian Cao

Keywords:
Additive manufacturing
Powder bed fusion
Residual stress
Surrogate model
Convolutional neural network

ABSTRACT

Over the past decade, the Laser Powder Bed Fusion (LPBF) process has been widely used in the fabrication of industrial parts with advanced functions. It is known that the complex thermal processing of the material during the LPBF process has a significant influence on product quality. While high fidelity simulation models can account for the effects of processing, they are generally too computationally expensive to be directly used in the design of components. Consequently, in this paper we propose a surrogate model for Simulation Models of the residual stress at the part-scale based on a Convolutional Neural Network (CNN) with a 3D U-Net architecture. In order to model the wide range of geometries that can arise during the design process, we developed a feature-based approach in which we trained our CNN on combinations of three basic types of geometric features: circular struts, square struts, and walls. Data augmentation was utilized to account for orientation invariance. Several benchmarks were designed to test the performance of the surrogate model. Results demonstrated that a CNN with a 3D U-Net architecture can accurately predict the residual stress for the features designed. The average training and testing errors are 5.3% and 6.6%, respectively. Prediction performance for the benchmark parts led to validation errors of 14.4%– 28.3% due to their complex geometries. Nevertheless, this strategy led to a significant reduction in runtime, demonstrating that the proposed feature-based surrogate model has the potential to replace high fidelity process simulations for the design of practical engineering parts manufactured using LPBF.

1. Introduction

The Laser Powder Bed Fusion (LPBF) process can fabricate metallic parts with intricate geometries (Gibson et al., 2021). The process provides a high level of design freedom for designers to improve the functional performance of products. Design for Additive Manufacturing (DFAM) was proposed by a lot of researchers in the last decade to exploit the capability of LPBF (Tang and Zhao, 2016). It was found that the in-process parameters may have significant influence on the performance of the product (Bang et al., 2021). To consider the manufacturing influence in the design stage, DFAM has developed from product design to product-process co-design.

The physics of the LPBF process have been studied for decades to understand material process-structure-property relationships (Yan et al., 2018), which is important to the product-process co-design method. Finite Element Analysis (FEA) is widely used to analyze the thermal and mechanical property during the LPBF process (Panda and Sahoo, 2019). However, a high-fidelity FEA model is computationally heavy due to the complex physical phenomena in the LPBF process. It may take hours or days to complete a single run, which is not ideal for the design and optimization process. Machine Learning (ML) has been used to provide cost-efficient approaches to analyze the LPBF process (Sing et al., 2021). It has been used for process parameter optimization and in situ monitoring. Simulation and experimental observations were analyzed by a Gaussian Process (GP)-based ML model to predict the remelted depth of single tracks as a function of laser powder and laser scan speed (Meng and Zhang, 2020). Similar technique was used to build a relationship between process parameters and relative density to discover the

* Corresponding author at: Department of Mechanical Engineering, University of Colorado Denver, Denver, CO, USA.
E-mail address: guoying.dong@ucdenver.edu (G. Dong).

https://doi.org/10.1016/j.jmatprotec.2022.117541
Received 3 October 2021; Received in revised form 22 December 2021; Accepted 26 February 2022
Available online 3 March 2022
0924-0136/

optimized parameter window (Liu et al., 2020). Physics-informed ML approaches can significantly improve the in situ quality control of the LPBF process as compared to pure data-driven ML algorithms (Gaikwad et al., 2020). Physical effect of the process parameter can be used to model the process-property relationship, instead of directly using process parameters (Liu et al., 2021). As for process monitoring, ML was used to make decisions and classifications in the closed-loop feedback control system of the PBF process (Amini and Chang 2018). A semi-supervised ML algorithm was introduced to detect the printing faults automatically by using the data from builds with both certified and unknown quality (Okaro et al., 2019). This approach is efficient especially when part certification is costly and time-consuming.

ML has also been widely used to develop surrogate models to accelerate the runtime speed of the in-process simulation with little sacrifice in accuracy (Kouraytem et al., 2021). Data-driven methods and artificial intelligence have been used to build surrogate models to predict the thermal field, microstructure, material properties, etc. Nath et al. (2018) used a Gaussian Process Regression (GPR) based surrogate models to quantify the uncertainty in the grain size distribution of the microstructure to replace the expensive FEA model. The GPR approach has also been used to build process planning surrogate models, which can predicts melt pool depth (Tapia et al., 2018) or the temperature history (Kumar et al., 2021) in single-track experiments given a laser power, scan speed, and laser beam size combination. Li et al. (Li et al., 2018) proposed a physically-based and data-driven approach to predict the thermal history of the LPBF process. FEA modeling results were sampled to establish a surrogate model using GPR method and Bayesian calibration. It showed that the proposed model can predict the thermal field quickly with different process parameters and geometry designs. Besides GPR, other data-driven approaches such as polynomial chaos expansion (PCE) and principal component analysis (PCA) were used to build surrogate models to simulate multi-physics field in the LPBF process (Hosseini et al., 2021) and reduce the dimensions in model inputs and field output (Vohra et al., 2020). Roy and Wodo (2020) proposed a neural network (NN) based surrogate model to predict thermal history of the LPBF process. They directly translated a part's manufacturing G-code into a set of features instead of using the part geometry. Results showed that the model can achieve high accuracy (below 5%) with a low computational cost (0.034 s).

However, most existing research has focused on simple geometries. Such surrogate models have only demonstrated on very limited cases such as single track or multi-track scans with simple bulky shapes, which is not sufficient for the co-design of product and process. During the design process, the shape of the part will be changed to achieve better functional performance. A more general surrogate model that can be used for different shapes and structures is needed. In this work, a feature-based surrogate model was developed to predict part properties fabricated by the LPBF process. Basic features such as struts and walls were used to generate a large set of geometries to comprise the geometry bank. A high fidelity LPBF process Full Order Model (FOM) was used to analyze the geometry bank and generate the training data (Mikula et al., 2021). A 3D U-Net Convolutional Neural Network (CNN) was used to build the surrogate model because of its good capability of learning abstract features directly from voxelized geometries without requiring explicit parameterization. (Çiçek et al., 2016). With the U-Net architecture design, our model produces output value(s) to each input voxel in 3D, allowing it to predict the entire distribution of part properties. Moreover, the U-Net architecture design with skip connections is efficient to be trained. These characteristics make 3D U-Net CNN one of the most suitable ML method to develop feature-based surrogate model for complex geometries.

In Section 2, some background about the FOM and 3D U-Net are introduced. Then, the details of the surrogate model development are presented in Section 3. The performance of the surrogate model is discussed in Section 4 and conclusions are drawn in Section 5.

2. Background

2.1. LPBF full order model

Data-driven approaches to predict the thermal and mechanical property in the LPBF process are usually based on the data obtained from the simulation model. In this paper, the data used for feature-based surrogate model are generated by an in-house FOM of the LPBF process that was developed and validated by the authors (Mikula, 2021). It is composed of a voxel-based meshing capability, finite element solver, and a material model, which was implemented within a framework that allows batch job generation, execution, and post-processing. Implementation was such that all inputs were specified in a plain text file, where keywords were used to identify each type of input. The input file was read by an Application Programming Interface (API) executable, which was designed to communicate with a backend, which performed the actual numerical simulations. Such architecture had the advantage of decentralized computing, whereby the API could run on a low-end terminal hardware, while the backend was running on a high-end workstation. Central to the present discussion, the mechanical solver was based on the inherent strain method, whereby the inherent strain was the initial condition of each new added element as mesh layers where sequentially added. As boundary condition, the global bottom nodes were kept at zero displacement to model part's attachment to the baseplate.

In our calculations, we assumed the material to be nominally maraging steel with Young's modulus $E = 210GPa$, Poisson's ratio $\nu = 0.3$, yield stress $\sigma_y = 1500MPa$, hardening coefficient $h = 0$ and inherent strain $\boldsymbol{\varepsilon}^{inh} = -0.008$. The stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$ fields were derived from the displacement field $\boldsymbol{u}$ through the simple constitutive law:

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}^{el} = \boldsymbol{C} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{inh} - \boldsymbol{\varepsilon}^{p}\right) \tag{1}$$

where $\boldsymbol{\varepsilon}^{el}$ is the elastic strain, $\boldsymbol{\varepsilon}$ is the total strain tensor $\varepsilon_{ij} = 1/2\left(\partial u_i/\partial x_j + \partial u_j/\partial x_i\right)$ in which $\boldsymbol{u} = \left(u_x, u_y, u_z\right)$ is the displacement field vector and $\boldsymbol{x}$ is the position vector. Here, $\boldsymbol{C}$ is the tensor of elastic constants, the components of which depend on E and ν, and $\boldsymbol{\varepsilon}^{p}$is the plastic strain tensor.

In the backend, numerical simulations were performed as follows. The component was meshed using brick elements, where the Visualization Toolkit (VTK) (Schroeder, 2006) and CGAL (CGAL) were used to compute the elements from the part's STL file. The finite element solution was found by minimizing the total strain energy functional Ψ:

$$\Psi = \frac{1}{2}\int_{\Omega}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{p} - \boldsymbol{\varepsilon}^{inh}\right) : \boldsymbol{C} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{p} - \boldsymbol{\varepsilon}^{inh}\right)\mathrm{d}\Omega \tag{2}$$

Minimization of this functional (within the FE framework) leads to a system of linear equations $\boldsymbol{Ku} = \boldsymbol{F}$ at every load increment, where $\boldsymbol{K}$ is the global stiffness matrix and $\boldsymbol{F}$ is the global vector of nodal forces. Implementation was carried out using the Trilinos library ('The Trilinos Project') in C++ language. The results from FOM were saved as VTK files. The data was then used to train the 3D U-Net CNN model.

2.2. 3D U-net convolutional neural network

By mimicking the vision processing in biological brain, recent methods based on deep CNNs have outperformed classical machine learning methods in many visual recognition tasks (LeCun et al., 2015) such as object classification (Krizhevsky et al., 2012), object segmentation (Long et al., 2015), medical image analysis (Yamashita et al., 2018), etc. One of the major advantages of CNNs is that they are capable to learn abstract features directly from raw inputs, whereas classical machine learning methods heavily rely on hand-engineered features and their quality. Therefore, CNNs are particularly suitable for tasks where inputs are difficult to parameterize, such as complex geometries in 3D volumes. CNNs can be directly applied to the 3D voxel representation

without the need of additional parameterization of such geometries. In comparison to more traditional multilayer perceptron (MLP) networks, CNNs have the advantage of being able to exploit spatial locality, which potentially helps CNNs to achieve better generalization. The weights sharing property of CNNs also dramatically reduces the number training parameters, allowing for training more powerful networks for complex problems.

The U-Net structure, originally developed for biomedical image segmentation (Ronneberger et al., 2015), has become a promising choice of CNN architecture for the *surrogate modeling task* where the desired output is supposed to contain value(s) assigned to each input pixel or voxel in 3D (Yao et al., 2018). Beyond medical applications, it has also been successfully applied to other problems such as fluid flow prediction (Thuerey et al., 2020). Briefly, the U-Net, as shown in Fig. 7, comprises of separate contraction (encoding) and expansion (decoding) halves. The contraction network is made up of successive layers of convolution, nonlinear activation, and pooling operations. They are designed to extract more advanced features from the input and reduce the feature map sizes. The expansion network is symmetric but acts in the opposite direction of the contraction half, where the convolution and pooling operations are replaced by (de)convolution and up-sampling operations. These expansion layers gradually increase the resolution of the outputs from the advanced features extracted by the contraction network. The U-Net structure features skip connections from layers of equal resolution in the contraction half to its expansion counterpart. Skip connections allow the network to propagate localization information to later layers, and back propagate the loss gradients to earlier layers more easily. The final output has the same resolution as the input, and yields a U-shaped architecture.

3. Methodology

The proposed surrogate model was developed by training a 3D U-Net CNN. The training data were obtained by running the FOM on STL models of parts consisting of the three types of features described earlier (thin walls, circular struts, square struts) that comprise the geometry bank. The FOM ran all the samples in the geometry bank and exports VTK files that contain the resulting physical fields from all samples. The residual stress field for each sample was represented as a voxel model, which was used as the training data in the CNN. But it should be noted that the proposed model can also work with other physical fields such as temperatures, thermal strains, displacements, and other material properties.

3.1. Feature design

The features that were studied in this research are combinations of struts with circular and square cross-sections and rectangular walls. Because intersections between features can cause stress concentrations, the design of parts for the geometry bank considered intersections between struts and walls. FreeCAD (FreeCAD, 2021) was used to generate the STL of the features. A Python script was developed to automatically create parametric features with defined parameter values. Due to computational cost, the target input size for the CNN model was a $64 \times 64 \times 64$ matrix. Generated geometries were designed to be contained within a 50 mm bounding box in x-y plane so that they do not exceed the target input size after data augmentation operations, which will be introduced in Section 3.3.

For the strut features, feature size was from 1 mm to 10 mm with 1 mm increments. Strut orientation was varied from 0° to 90° with 10° increment. For one type of feature part, a strut was intersected with a 5 mm thick wall and a vertical strut. For the wall features, the length of the wall length was chosen as 50 mm and its thickness was varied from 1 mm to 10 mm with 1 mm increments. The orientation angle was also from 0° to 90° with 10° increment. In the second type of feature part, a wall feature was intersected with two vertical walls on each side. The objective of the feature design is that the surrogate model can predict the physical field for features with different sizes, orientations, shapes, and intersections. All parts of the first type with circular struts are shown in Fig. 1 as an example. There are 110 parts for circular struts, square struts, and walls, respectively.

Some additional single strut and wall features were generated to enrich the geometry bank. The purpose is to consider the "gap" between strut and wall features. If two features are close to each other, the CNN may neglect the gap between features and consider them as one feature. These additional geometries were intended to train the CNN to recognize gaps between features. Firstly, single square strut and wall features were generated. Strut size and wall thickness was varied from 1 mm to 10 mm with 1 mm increment while heights were varied from 10 mm to 50 mm with 10 mm increment. For each size or thickness, 5 struts or walls, with different heights, were grouped together randomly in a row with gap sizes of 1–3 mm between them. The total number of parts with "gap" (grouped struts and walls) was 90. An example of the parts with "gap" for square struts is shown in Fig. 2.

Furthermore, four benchmark parts were also designed to evaluate the generalization of the CNN model, as shown in Fig. 3. Two contained combinations of the three feature types used for training. The other two were an original and an optimized design of an engineering bracket.

3.2. Data generation

Data for training the CNN model was obtained through a python script API that writes configuration files and executes them with the FOM through its API. This approach ensures consistency of the dataset and the ability to check the parameters and run status. The FOM was executed on a high performance workstation with two Intel(R) Xeon(R) Gold 6230 CPU@2.10 GHz chip using 32 cores and 512 GB memory. Consequently, we selected 1 mm cubic voxel resolution and ensured that all parts in the database fit within the $64 \times 64 \times 64$ voxels matrix. With this setting, the numbers of elements in the FOM are typically $O(10^4)$ since the solid parts only occupy small portions of the computational domain. Full order simulations with the FOM required approximately 1 h for each part in the geometry bank while benchmark geometries took up to 4 h due to their larger density. Total simulation time for over 420 geometries amounted to 16,000 CPU hours or 3 weeks on the workstation used. Fig. 4 illustrates the mesh as well snapshots of the layer-by-layer simulation. From the figure, the typical evolution of the stress field during fabrication can be seen.

3.3. Data augmentation

Data augmentation is important to improve the performance of the CNN model for test parts that are rotated or translated relative to training models. That is, the CNN performance should be invariant to rotations or translations of part models. Some object classification methods utilize shape descriptors that are rotation and translation invariant (Zhang et al., 2019). Usage of these methods would eliminate the need for the data augmentation approach presented here and will be investigated in future work. In this study, each pair of input and output parts was rotated by the same angle to generate augmented data. The angle was chosen randomly between 0 and 360 degrees. Initially, all the parts were generated in the same direction.

The parts were only rotated about the z-axis because various feature orientations were incorporated into the geometry bank already. Each feature model was placed at the bottom center of the 3D matrix. The top view of some of the rotated features is shown in Fig. 5(a). Red color indicates the features, while purple indicates the space outside the feature.

The CNN prediction outside the feature is irrelevant in the model fitting. Because only the value inside the feature are important in this research. If the loss outside the feature is considered by the CNN model, the training parameters will not be fully optimized to minimize the loss

Fig. 1. All the circular strut features designed by FreeCAD and Python script.

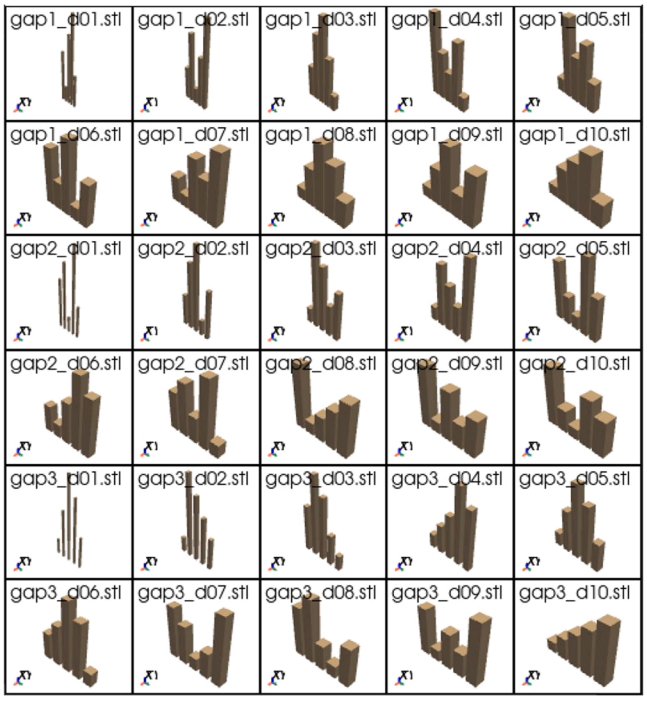

Fig. 2. Examples for square strut parts with "gap". Struts with the same thickness are randomly grouped together with a gap size from 1 mm to 3 mm.

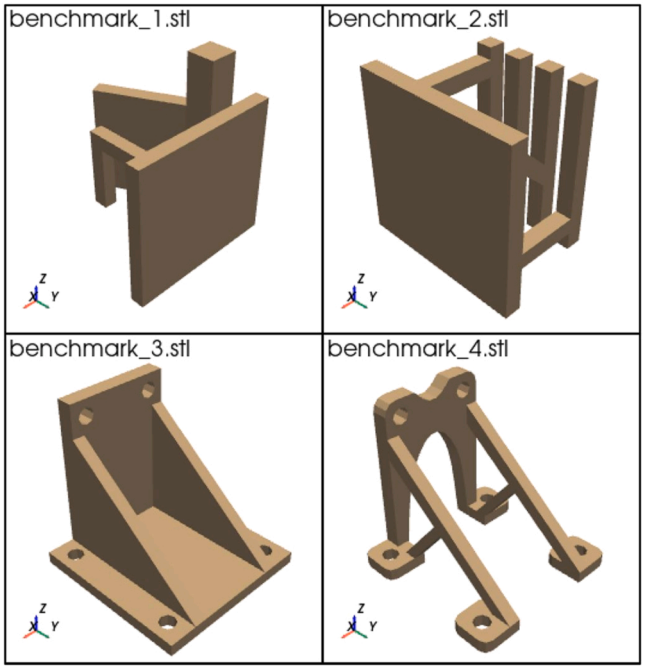

Fig. 3. Benchmarks are designed with multiple features to evaluate the performance of the surrogate model.

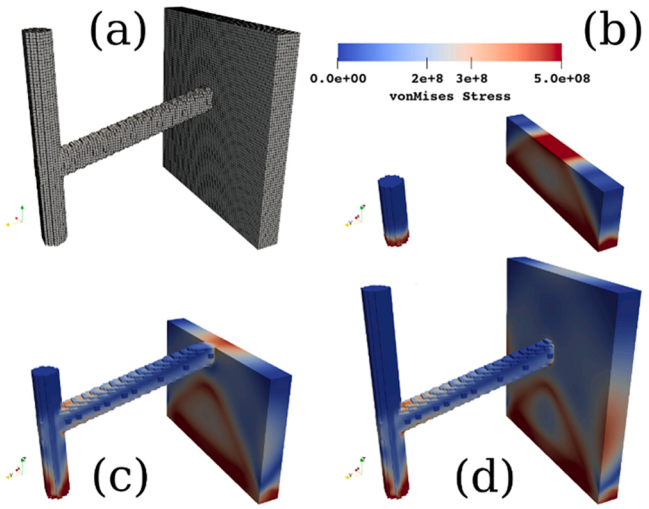

Fig. 4. Example of individual ground truth calculation of a strut part. (a) generated mesh; (b)-(d) progressive evolution of Von Mises stress field during build, captured at 33% ((b), 66% (c) and final stage during the layer-by-layer build.

on the feature. Thus, a weight matrix is needed to filter out the loss outside the feature. The input matrix of the CNN model was also used as a weight matrix because the voxel value is 1 when it is inside the feature, while the value is 0 when the voxel is outside the feature. The loss matrix was multiplied by the weight matrix to eliminate the loss value outside the feature. The training parameters were optimized only to minimize the loss inside the feature.

Furthermore, in the feature design process, each feature was intersected with a wall that has a constant size. The objective is to train the CNN model with intersection features. The intersection wall itself is not that important in the training process. Therefore, the weight matrix was adjusted to control the weight of the intersection wall. A weight adjustment matrix was generated automatically by locating the intersection wall feature in the input matrix. This matrix was rotated by the same angle as the corresponding input matrix as shown in Fig. 4(b). The wall feature in the weight adjustment matrix overlaps the intersection wall in the input data. A weight adjustment factor was used to control the importance of the intersection wall. The total weight matrix is calculated by Eq.(3).

$$W = X - (1 - f) * W_a \tag{3}$$

where X is the input matrix, W_a is the weight adjustment matrix of the intersection wall. f is the weight adjustment factor. If $f = 1$, the weight of the intersection wall is the same as other features. If $f = 0$, it means that the prediction of the intersection wall will not be considered in the CNN optimization process. In this paper, f was set to 0.5 because it can balance the importance of the strut and wall features. The total weight matrix W is used in the CNN model fitting function.

The augmented data were compared to the ground truth (FOM results) to validate the accuracy of the rotation. The rotated input matrix was imported to the FOM as a mesh to get the ground truth of the rotated feature. The ground truth was compared with the corresponding stress field of the rotated output matrix voxel by voxel. The value of the ground truth and the error are shown in Fig. 6. The total relative error, defined by Eq.(4) is 3.85%. It proved that the data augmentation method did not reduce the accuracy of the output result and these data can be used to improve the performance of the CNN model.

3.4. CNN architecture

As motivated in Section 2.2, the 3D U-Net CNN was selected for use in this surrogate modeling task, where the input and output have identical spatial dimensions of 64 × 64 × 64. The architecture used in the present study is illustrated in Fig. 7. The contraction network gradually maps the input into 8, 32, 128, and 256 feature maps. A max pooling operation is applied to the feature maps after every 2 convolutional layers, to reduce the size of feature maps by a half in each dimension. At the bottom of the U-Net, 256 feature maps with spatial dimension of 8 × 8 × 8 are created and passed to the expansion network, where up-sampling and the skip connections are applied after every alternate convolutional layers, to gradually map these feature maps to 128, 32, and 8 outputs with increasing resolution. All convolutional layers, except the final one, use a 3D convolutional kernel of size 3 × 3 × 3, batch normalization, and a ReLU activation component.

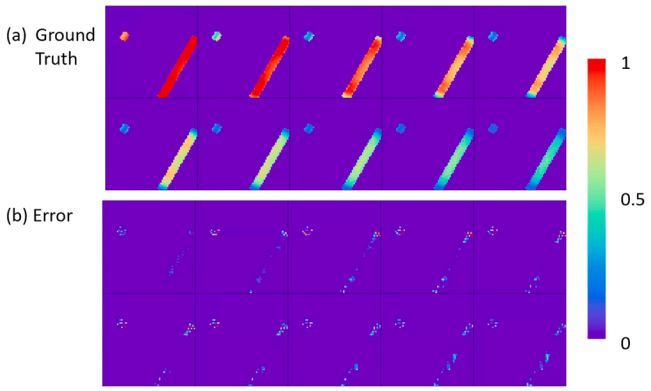

Fig. 6. (a) The ground truth of the rotated feature and (b) the error between the ground truth and the rotated output.

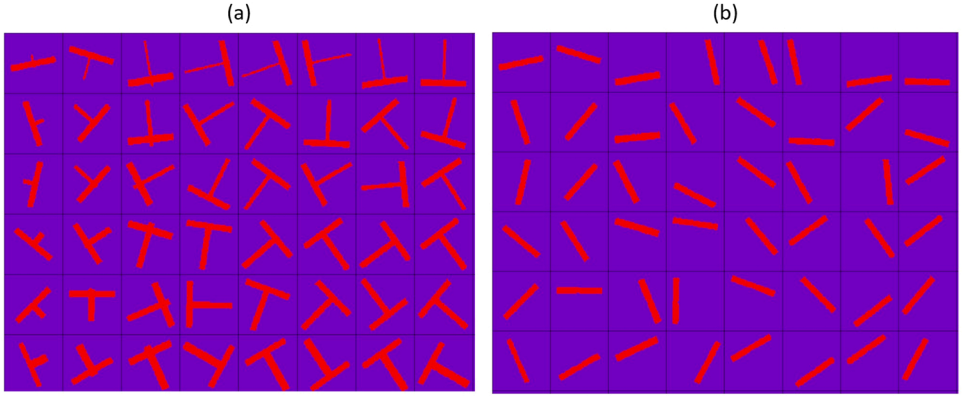

Fig. 5. Top views of (a) input matrices after data augmentation and (b) weight adjustment matrices.

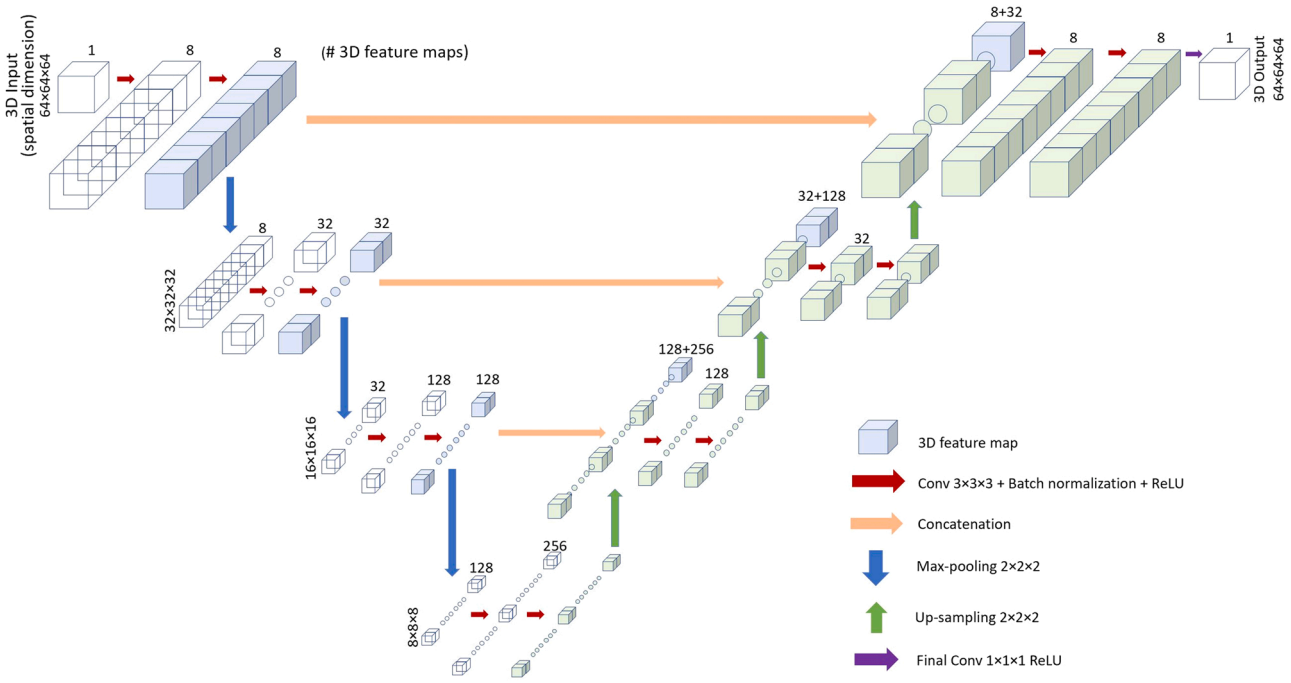

Fig. 7. Schematic of the 3D U-Net architecture as implemented in this study.

All the inputs to convolutional layers are padded with zeros such that the outputs have the same spatial dimension as the input after the convolution. The final convolutional layer has a kernel of size 1×1, a batch normalization, and a ReLU activation, to produce a single output with a spatial dimension of $64 \times 64 \times 64$. The ReLU activation in the final layer ensures the outputs – von Mises stress predictions – are non-negative. The network weights are initialized with the Xavier method (Glorot and Bengio, 2010). The U-Net model was created using the Keras API as packaged with TensorFlow2.5 (Abadi et al., 2016), and has slightly above 5 million trainable parameters.

To identify the optimal parameters for the network, several variants of U-Net design have been explored. The impact of the kernel size (e.g., $3 \times 3 \times 3$, $4 \times 4 \times 4$, $5 \times 5 \times 5$), activation (tanh, leaky ReLU, etc.), initialization scheme (Xavier, He), and the presence of batch normalization and skip connections were studied. The final architecture was chosen based on the accuracy, smoothness of the solution, and training efficiency. The tested and selected hyperparameters are shown in Table 1. With the selected hyperparameters, the overall error was lower than the alternatives on the training data, test data, gap features and the benchmarks. Based on our experiences, a larger network by increasing the network depth and the filter size may be beneficial, however enlargement may have a diminishing impact on both training and validation accuracy while being much more computationally expensive to train.

Similarly, a brief parametric study was conducted to search for an appropriate training setting. The loss function used for training is defined as the mean absolute error (MAE) between the ground truth of the output Y_t and the U-Net prediction Y_p, in addition to an L1-regularizer with regularizer weight = 1e-6 to prevent over-fitting. The state-of-the-art ADAM optimizer (Kingma and Ba, 2014) was used for training the U-Net, with the termination criteria set as 500 epochs, or when the validation loss falls below 1e-4. A mini batch size of 8 was chosen. Prior literature in the machine learning domain typically identified the learning rate as one of the most important hyper-parameters in deep neural network training (Bengio, 2012). To improve the efficiency and effectiveness of training, a decaying learning rate with an initial value of 5e-4 was used in the present study.

Table 1
The hyperparameters that are tested in the CNN architecture design and the selected value based on error evaluation.

Hyperparameters	Tested Value	Selected Value
Max. filter size	64, 128, 256	256
Kernel size	3, 4, 5	3
Activation function	'ReLU', 'leaky ReLU', 'tanh'	'ReLU'
Initializer	'Xavier', 'He'	'Xavier'
Loss function	'MSE', 'MAE'	'MAE'
Regularizer weight	$1e^{-4}$, $1e^{-5}$, $1e^{-6}$	$1e^{-6}$
Initial learning rate	$5e^{-3}$, $1e^{-3}$, $5e^{-4}$, $1e^{-4}$	$5e^{-4}$
Epoch	100, 200, 500	500
Batch size	8, 16, 32, 64	8

4. Results and discussions

4.1. Training and testing results

Six geometries were randomly selected from the geometry bank and were reserved for testing. The details of the test data are shown in Table 2. Of the remaining geometries, 90% were used in the training and 10% were used as validation data. The loss value in the background was ignored by using the weight matrix calculated by Eq.(3). The CNN model was trained with GPU acceleration. The training history in Fig. 8 shows that the CNN model converged within 500 epochs. The relative error function as shown in Eq.(4).

$$E = \sum\left(\frac{\|Y_t - Y_p\|}{\|Y_t\|} \odot X\right) \times 100\% \tag{4}$$

Table 2
The details of the data that are reserved for testing.

Test ID	Feature Type	Size	Inclined Angle
1	Square Strut	3 mm	-30°
2	Square Strut	10 mm	75°
3	Circular Strut	6 mm	15°
4	Circular Strut	8 mm	60°
5	Circular Strut	10 mm	75°
6	Wall	6 mm	15°

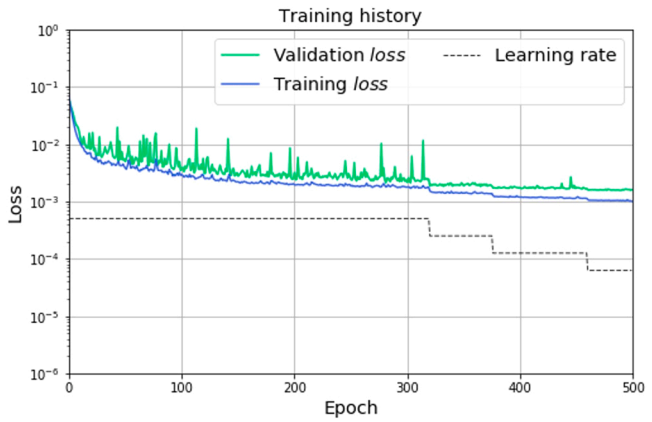

Fig. 8. The loss history during the training of the CNN model.

was used to evaluate the performance of the CNN model, where X is the input matrix, Y_t is the target output matrix, Y_p is the CNN prediction output matrix. The relative error calculated between Y_t and Y_p was multiplied by the input matrix X to ignore the background. The error inside the feature was kept (multiplied by 1 in the input matrix) while the error outside the feature was eliminated (multiplied by 0 in the input matrix). It was found that the relative error of most training data is from 2% to 9% except for some outliers. The average error of the training dataset is 5.28%.

The prediction results of six test geometries are shown in Fig. 9. Results of the first three test parts are shown in the top half of the figure. The three plots in the first row are the ground truth stress field, while the plots in the second row are the results predicted by the CNN model. The other three test part results are shown in the bottom half of the figure. Note that the stress field is normalized so the values in the plots are between 0 and 1. The test data are also rotated by a random angle in the data augmentation process. The orientation of each test data is different. It is found that the prediction error of the test data is similar to the training data, from 5.6% to 8.0%. It shows that the CNN model can learn from the training features and predict results accurately on the testing features.

4.2. Rotation and translation invariance

As mentioned, prediction performance should be independent of part position and orientation. To demonstrate rotation and translation (shift) invariance, several tests were conducted. First, single square struts with 5 mm thickness and 50 mm height were tested. A single square strut in the bottom center of the 64 × 64 × 64 spatial domain was simulated by the FOM to obtain the ground truth. Then, this square strut was placed at a corner of the input matrix and shifted along the x and y axes voxel by voxel and predictions were made at each location. The resulting relative error plot is shown in Fig. 10(a). When the strut was placed near the boundary of the matrix, the prediction error was very large, over 30%. But when the strut was several voxels away from the boundary, the error decreased below 5%, which is similar to the error of the training data.

The second shift invariance test was conducted on wall features. A wall with 5 mm thickness and 50 mm length and height was simulated by the FOM to get the ground truth. It was also shifted from corner to corner and voxel by voxel. The resulting error plot is shown in Fig. 10(b). Similar to the strut, if the front or the back face of the wall was close to the boundary, the prediction error was as large as 12%. The error decreases to 2–4% when the wall is away from the boundary. But if the side face of the wall is close to the boundary, the error does not increase. It can be concluded that the CNN model has good shift invariance property, but features should not be close to the boundary of the input matrix.

Rotational invariance was also tested in the CNN model. 36 geometries were generated by rotating Test-1 geometry in Fig. 9 about its z axis in 10° increments. CNN predictions were compared to FOM results for each model and relative errors are shown in Fig. 11. Resulting errors were between 4% and 7%, which is similar to the training error, indicating that the use of data augmentation yields a CNN model that has good rotational invariance. However, if data augmentation is not used, CNN prediction accuracy is significantly degraded, as shown by the orange curve in Fig. 11. The reason is that, in the data augmentation process, the training data are rotated by a random angle. If there is not data augmentation and if all the features are in the same orientation, the CNN model will not have good rotational invariant property.

4.3. Feature with gaps

In the LPBF process, if two parts are not connected, even if the gap is small, there is no strong interaction between each other in the FOM. The physical property fields on each part are independent. However, due to the construction of the CNN, small gaps between features may not be considered significant, meaning that the features may be considered as one feature. This is indicated by the example in Fig. 12. In this experiment, 5 struts were placed in the spatial domain with gap sizes from 0 to 8 mm. The first set of predictions was performed by a CNN model that was trained only with the strut and wall features (Fig. 1) but not the parts with “gap” (Fig. 2). These are the results shown in the first row of plots in Fig. 12. When the gap size is 0 or small (1 mm or 2 mm), the prediction result of the middle strut is higher than those on the ends. The CNN model considers these 5 struts as a wall when the gap is small.

To improve the performance of the CNN model on the parts with gaps, additional parts introduced in Section 3.1 were added into the geometry bank. The CNN model was retrained with the extended geometry bank and was used to generate a second set of predictions. Results are shown in the second row of Fig. 12. In this case, the predictions are different for no gap and small gaps, which means that the CNN model can recognize the gaps and generate predictions accordingly.

To investigate these results further, the relative error between the middle strut and a single strut was calculated to quantify the influence of gaps. The prediction result of a single strut is shown in Fig. 13(a). This prediction is compared to those of the middle struts with gap size from 1 mm to 10 mm. The relative error between them was calculated by Eq. (4) and is shown in Fig. 13(b). Results show that, if the parts with “gap” are not added, the error decreases with an increase of gap size. When the gap is 1 mm, the error is almost 140%. When then gap is 10 mm, the error drops to 15%. The prediction result on the middle strut in the second row of Fig. 12 was also compared to the single strut prediction. In this case, where parts with “gap” were added, the error is only 1%− 5%. It indicates that the CNN model can learn from the added parts and recognize the gap between the struts very well. The predicted stress on the strut is not influenced by other struts close to it. As demonstrated, the inclusion of parts with “gap” can significantly improve the performance of the CNN model.

4.4. Benchmark result

The benchmark parts were designed to evaluate the performance of the CNN model on more complex features. The prediction results on the benchmarks are shown in Fig. 14. The error map shows the difference between the prediction and the ground truth. Benchmark 1&2 were designed with multiple strut or wall features. Benchmark 3&4 are engineering brackets with original and optimized designs. Results demonstrate that the CNN model can predict accurately the stress distributions on the benchmarks, even though the overall geometries of benchmarks are much different from the training features. Further, the CNN model can predict the high stress areas on the benchmark parts. We expected this since most features in the benchmark parts are covered by the training data.

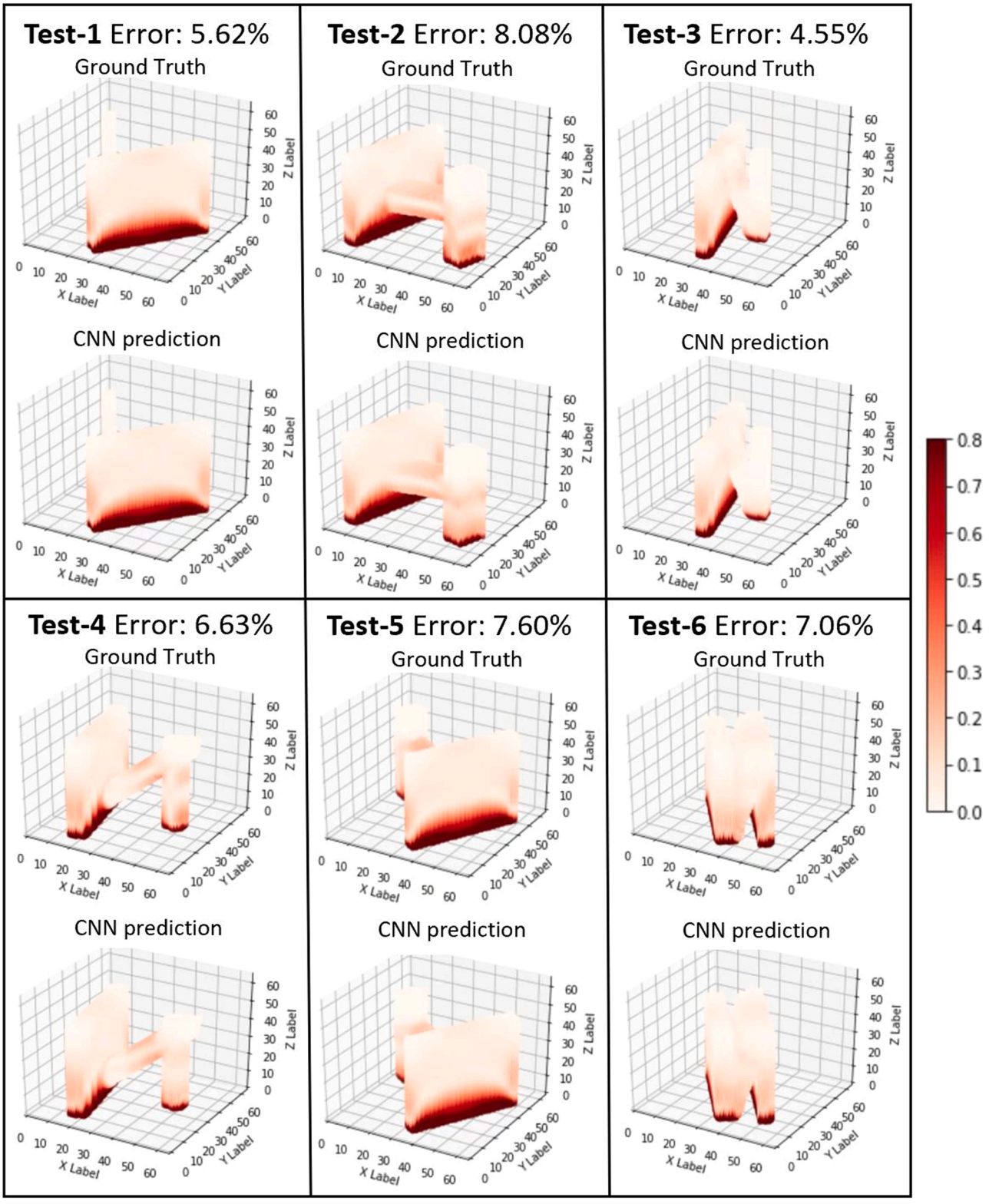

Fig. 9. Comparison of the stress field between target and the prediction.

In summary, with the usage of data augmentation and gap features in training, CNN prediction accuracy can be significantly improved. The prediction errors on the benchmarks with different training set are summarized in Table 3. If the CNN model is trained without gap features or data augmentation, prediction errors can reach 60% and above. The data augmentation and gap features can reduce the average prediction error from 37.7% to 21.3%. It was also found that the relative errors on the benchmark parts are larger than that on the test parts. One reason is that the test data were selected from the geometry bank, so the similarity between the training and test data were higher than that between the training data and the benchmarks. For benchmark parts 2 and 4, error were mainly concentrated near the intersection of features. Therefore, to further improve the performance of the CNN model, more features with intersections should be designed and added into the geometry bank.

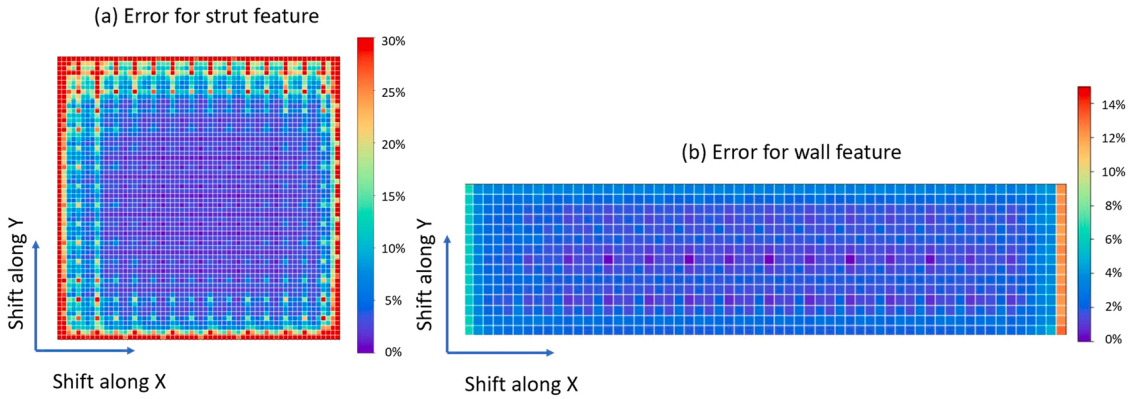

Fig. 10. Error comparison between shifted feature and centered feature. One grid means that the feature is shift one voxel along x or y axis.

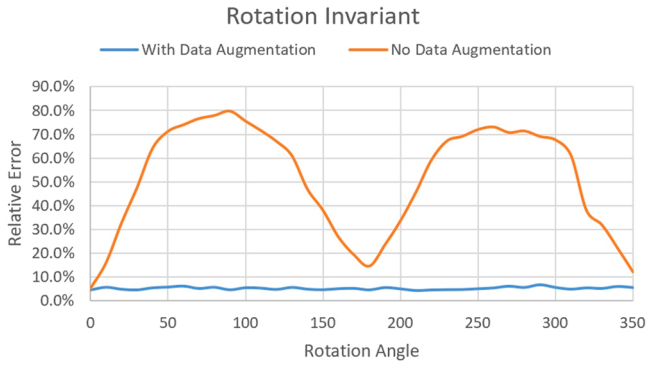

Fig. 11. The error of the Test-1 part rotated from 0° to 350°.

5. Conclusion

In this research, a feature-based surrogate model was proposed to predict the residual stress of parts fabricated through the LPBF process. Models with strut and wall features were generated to train the surrogate model. Data augmentation with rotated features and gap features was used to improve the performance of the surrogate model. A 3D U-Net CNN model was designed to be the core of the surrogate model.

Results demonstrated that the CNN model can accurately predict the stress field in LPBF fabricated parts. The CNN model has good shift and rotational invariance properties if the training data are appropriately augmented. It was also found that the CNN model had difficulty in the prediction of features with small gaps. By adding gap features into the geometry bank, prediction accuracy on feature models with gaps was significantly improved. Some benchmark parts were designed to test the overall performance of the surrogate model on more complex, industry-relevant parts. With the increase of part complexity, prediction accuracy on the benchmarks was lower than for the testing data. However, the overall stress distributions and locations of stress concentrations were predicted accurately.

Future work will focus on improving prediction accuracy by modifying the 3D CNN architecture and by adding more training data. Usage of rotationally invariant shape descriptors will also be investigated to try to avoid the need for data augmentation. Application of the CNN to a wider variety of more complex parts will also be pursued.

CRediT authorship contribution statement

Guoying Dong: Methodology, Surrogate model development, Writing – original draft preparation, **Jian Cheng Wong:** Surrogate model development, Writing – original draft preparation, **Lucas Lestandi:** Data generation, Writing – original draft preparation., **Jakub Mikula:** Full-order-model development, **Guglielmo Vastola:** Full-order-model development, Writing – original draft preparation, **Mark Hyunpong Jhon:** Writing – review and editing, **My Ha Dao:** Writing – review and editing, **Umesh Kizhakkinan:** Full-order-model development,**Clive Stanley Ford:** Writing – review and editing, Supervision, **David William Rosen:** Methodology, Writing – review and editing, Supervision.

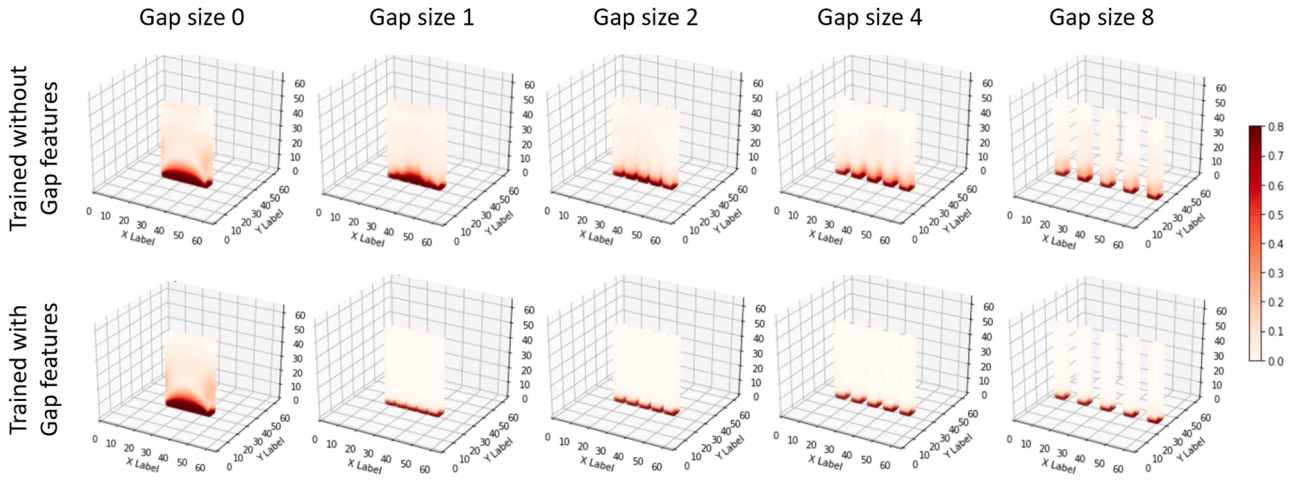

Fig. 12. Strut with gaps to test the influence of the gap on the prediction accuracy of the CNN model.

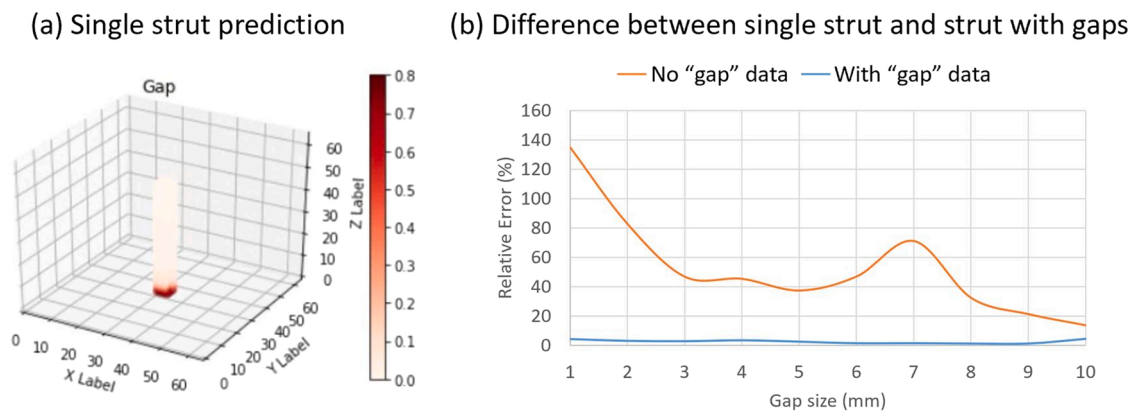

Fig. 13. (a) the CNN prediction result of a single strut, (b) the error between the center strut in the five struts with gaps and the single strut.

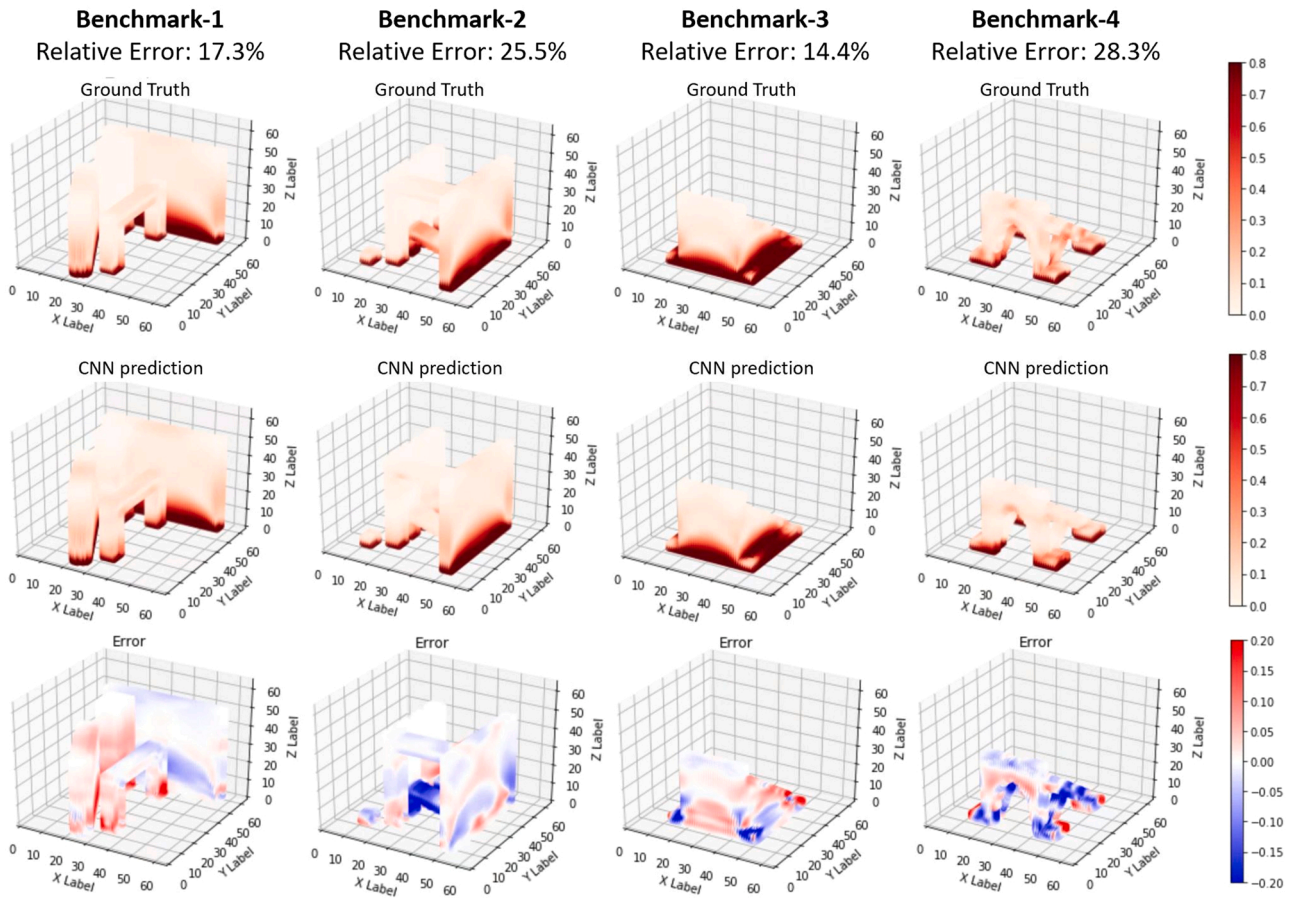

Fig. 14. The CNN prediction results of the benchmarks, the first row is the targets of the benchmarks, the second row is the CNN predictions, the third row is the error between the prediction and the target.

Table 3
The CNN prediction errors on the benchmarks parts with different training dataset.

Training Dataset		1	2	3	4
Data Processing	Data Augmentation	No	No	Yes	Yes
	Add Gap features	No	Yes	No	Yes
Error	Benchmark-1	38.3%	35.1%	20.9%	17.3%
	Benchmark-2	54.6%	62.6%	25.5%	25.5%
	Benchmark-3	31.9%	27.5%	18.8%	14.4%
	Benchmark-4	26.0%	31.4%	26.3%	28.3%
	Average	37.7%	39.2%	22.9%	21.3%

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.

Acknowledgment

This research is supported by Agency for Science, Technology and Research (A*STAR), Singapore, under its IAF-PP project: Industrial Digital Design and Additive Manufacturing Workflows (IDDAMW)

[Award No. A19E10097].

Appendix

The CNN model, training data and test data used in this research can be downloaded from:
https://data.mendeley.com/datasets/8rm9d4ykbt/1.

References

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. Tensorflow: A system for large-scale machine learning. In: Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16, pp. 265–283.

Amini, M., Chang, S.I., 2018. MLCPM: a process monitoring framework for 3D metal printing in industrial scale. Comput. Ind. Eng. 124, 322–330.

Bang, Gyung, Bae, Won Rae, Kim, Hyo Kyu, Kim, Hyung-Ki, Park, Gun Hee, Kim, Soong-Keun, Hyun, Ohyung, Kwon, Kim, Hyung Giun, 2021. Effect of process parameters for selective laser melting with SUS316L on mechanical and microstructural properties with variation in chemical composition. Mater. Des. 197, 109221.

Bengio, Yoshua, 2012. Practical recommendations for gradient-based training of deep architectures. Neural Networks: Tricks of the Trade. Springer.

CGAL. 'Computational Geometry Algorithms Library, ⟨https://www.cgal.org⟩.

Çiçek, Özgün, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, Olaf Ronneberger, 2016. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, edited by Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal and William Wells, 424–432. Cham: Springer International Publishing.

'FreeCAD'. 2021. ⟨https://www.freecadweb.org/⟩.

Gaikwad, Aniruddha, Giera, Brian, Guss, Gabriel M., Forien, Jean-Baptiste, Matthews, Manyalibo J., Rao, Prahalada, 2020. Heterogeneous sensing and scientific machine learning for quality assurance in laser powder bed fusion – a single-track study. Addit. Manuf. 36, 101659.

Gibson, Ian, David, W.Rosen, Stucker, Brent, Mahyar Khorasani, 2021. Additive Manufacturing Technologies. Springer.

Glorot, Xavier, Yoshua Bengio, 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–56. JMLR Workshop and Conference Proceedings.

Hosseini, Ehsan, Ghanbari, P.Gh, Keller, F., Marelli, S., Mazza, Edoardo, 2021. Deploying artificial intelligence for component-scale multi-physical field simulation of metal additive manufacturing. In: Meboldt, Mirko, Klahn, Christoph (Eds.), Industrializing Additive Manufacturing, 268–76. Springer International Publishing, Cham.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: a method for stochastic optimization. arXiv Prepr. arXiv 1412, 6980.

Kouraytem, Nadia, Xuxiao, Li, Wenda, Tan, Branden, Kappes, Spear, Ashley D., 2021. Modeling process–structure–property relationships in metal additive manufacturing: a review on physics-driven versus data-driven approaches. J. Phys. Mater. 4, 032002.

Krizhevsky, Alex, Ilya, Sutskever, Geoffrey, Hinton, E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105.

Kumar, Anandan, Hemnath, Kumaraguru, Senthilkumaran, Paul, C.P., Bindra, K.S., 2021. Faster temperature prediction in the powder bed fusion process through the development of a surrogate model. Opt. Laser Technol. 141, 107122.

LeCun, Yann, Bengio, Yoshua, Geoffrey, Hinton, 2015. Deep learning. Nature 521, 436–444.

Li, Jingran, Ran, Jin, Yu, Hang Z., 2018. Integration of physically-based and data-driven approaches for thermal field prediction in additive manufacturing. Mater. Des. 139, 473–485.

Liu, Qian, Wu, Hongkun, Paul, Moses J., He, Peidong, Peng, Zhongxiao, Gludovatz, Bernd, Kruzic, Jamie J., Wang, Chun H., Li, Xiaopeng, 2020. Machine-learning assisted laser powder bed fusion process optimization for AlSi10Mg: new microstructure description indices and fracture mechanisms. Acta Mater. 201, 316–328.

Liu, Rui, Liu, Sen, Zhang, Xiaoli, 2021. A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing'. Int. J. Adv. Manuf. Technol. 113, 1943–1958.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Meng, Lingbin, Zhang, Jing, 2020. Process design of laser powder bed fusion of stainless steel using a Gaussian process-based machine learning model. JOM 72, 420–428.

Mikula, Jakub, Ahluwalia, Rajeev, Laskowski, Robert, Wang, Kun, Vastola, Guglielmo, Zhang, Yong-Wei, 2021. Modelling the influence of process parameters on precipitate formation in powder-bed fusion additive manufacturing of IN718. Mater. Des. 207, 109851.

Mikula, Jakub. 2021. 'Hitachi Metals Technical Review 37', 10.

Nath, Paromita, Zhen Hu, Sankaran Mahadevan. 2018. Modeling and uncertainty quantification of material properties in additive manufacturing. In: Proceedings of the 2018 AIAA Non-Deterministic Approaches Conference (American Institute of Aeronautics and Astronautics).

Okaro, Ikenna A., Jayasinghe, Sarini, Sutcliffe, Chris, Black, Kate, Paoletti, Paolo, Green, Peter L., 2019. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. Addit. Manuf. 27, 42–53.

Panda, B.K., Sahoo, S., 2019. Thermo-mechanical modeling and validation of stress field during laser powder bed fusion of AlSi10Mg built part. Results Phys. 12, 1372–1381.

Ronneberger, Olaf, Fischer, Philipp, Brox, Thomas, 2015. U-net: convolutional networks for biomedical image segmentation. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Roy, Mriganka, Wodo, Olga, 2020. Data-driven modeling of thermal history in additive manufacturing. Addit. Manuf. 32, 101017.

Schroeder, Will, Martin, Ken, Lorensen, Bill, 2006. The Visualization Toolkit, fourth ed.

Sing, S.L., Kuo, C.N., Shih, C.T., Ho, C.C., Chua, C.K., 2021. Perspectives of using machine learning in laser powder bed fusion for metal additive manufacturing. Virtual Phys. Prototyp. 16, 372–386.

Tang, Yunlong, Zhao, Yaoyao Fiona, 2016. A survey of the design methods for additive manufacturing to improve functional performance. Rapid Prototyp. J. 22, 569–590.

Tapia, Gustavo, Khairallah, Saad, Matthews, Manyalibo, King, Wayne E., Elwany, Alaa, 2018. Gaussian process-based surrogate modeling framework for process planning in laser powder-bed fusion additive manufacturing of 316L stainless steel. Int. J. Adv. Manuf. Technol. 94, 3591–3603.

The Trilinos Project, https://trilinos.github.io.

Thuerey, Nils, Weißenow, Konstantin, Prantl, Lukas, Hu, Xiangyu, 2020. Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. AIAA J. 58, 25–36.

Vohra, Manav, Nath, Paromita, Mahadevan, Sankaran, Lee, Yung-Tsun Tina, 2020. Fast surrogate modeling using dimensionality reduction in model inputs and field output: Application to additive manufacturing. Reliab. Eng. Syst. Saf. 201, 106986.

Yamashita, Rikiya, Nishio, Mizuho, Do, Richard Kinh Gian, Togashi, Kaori, 2018. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629.

Yan, Wentao, Lin, Stephen, Kafka, Orion L., Yu, Cheng, Liu, Zeliang, Lian, Yanping, Wolff, Sarah, Cao, Jian, Wagner, Gregory J., Wing Kam, Liu, 2018. Modeling process-structure-property relationships for additive manufacturing. Front. Mech. Eng. 13, 482–492.

Yao, Wei, Zeng, Zhigang, Lian, Cheng, Tang, Huiming, 2018. Pixel-wise regression using U-Net and its application on pansharpening. Neurocomputing 312, 364-71.

Zhang, Zhiyuan, Binh-Son Hua, David W. Rosen, Sai-Kit Yeung. 2019. Rotation invariant convolutions for 3d point clouds deep learning. In: Proceedings of the 2019 International Conference on 3D Vision (3DV), pp. 204–13. IEEE.