



Deep learning-based melt pool and porosity detection in components fabricated by laser powder bed fusion

Zhaochen Gu¹ · K. V. Mani Krishna^{2,3} · Mohammad Parsazadeh^{2,3} · Shashank Sharma^{2,3} · Aishwarya Manjunath¹ · Hang Tran¹ · Song Fu¹ · Narendra B. Dahotre^{2,3}

Received: 18 June 2023 / Accepted: 19 March 2024
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2024

Abstract

Microstructure analysis is a crucial aspect of additive manufacturing (AM) processes, as it offers valuable insights into material properties, defects, and quality of printed parts. Quantification of microstructural features such as melt pool dimensions and porosity can help optimize the process parameters by aiding the development of useful correlations of the processing conditions and resulting print quality and properties. However, detecting melt pool boundaries and porosity-related defects in cross-sectional microstructure samples presents significant challenges due to the complex nature of these features and inherent difficulties in acquiring their images with quality and quantity required for accurate and efficient detection. To address this, we propose a deep learning-based approach that leverages state-of-the-art backbone models (EfficientNet b7 and DenseNet 201) with various convolutional neural networks (U-Net, LinkNet, and FPN) using transfer learning techniques to automatically segment and detect the melt pools and porosity from AM microstructure images. The results demonstrate our ability to accurately identify and segment melt pools and porosity, even with limited set of training data. A comparative study of the performance of the different neural network architectures was done and it was found that the quantification of results (of microstructural features) from the employed set of networks are statistically comparable although the accuracy from the combination of U-Net with EfficientNet b7 backbone was highest.

Keywords Additive manufacturing · LPBF · Deep learning · Segmentation · Melt pools · Porosity

1 Introduction

Additive manufacturing (AM), or 3D printing, is an emerging technology that enables the creation of complex structures potentially quickly and cost-effectively, revolutionizing manufacturing processes. Its ability to produce intricate designs makes AM a pivotal technology for Industry 4.0 [1], with wide applications in the medical, aerospace, and transportation fields. The quality of AM printing is critical, directly affecting the final product's reliability and safety [1].

Laser powder bed fusion (L-PBF) is a well-established metal 3D printing technique that has gained widespread attention and recognition in recent years. LPBF utilizes high-powered lasers that allow selectively melt and fuse the metal powder to rapidly produce printed parts through a layer-by-layer approach. During the printing process, a thin layer of metal powder is first spread over a build plate. Then, the laser beam is used to scan the surface of the powder bed, selectively melting, and fusing the metal particles to form a solid track. This process is iteratively repeated in a track-by-track, and layer-by-layer manner until the entire part is complete, resulting in a 3D metal object with complex geometries, which would be infeasible to achieve with traditional manufacturing methods.

During the metal printing processes, the melted material experiences a range of complicated physical processes [1], such as material compaction, heat transfer, powder melting, evaporation, and solidification. The solidification can result in the evolution of microstructure, which can be influenced by melt pool dynamics [2]. There are various factors that can

✉ Narendra B. Dahotre
Narendra.Dahotre@unt.edu

¹ Department of Computer Science & Engineering, University of North Texas, 3940 N Elm street, Denton, TX 76207, USA

² Department of Material Science & Engineering, University of North Texas, 3940 N Elm street, Denton, TX 76207, USA

³ Center for Agile and Adaptive Additive Manufacturing, University of North Texas, 3940 N Elm Street, Denton, TX 76207, USA

impact the final quality of the printed parts. For instance, the size and morphology of the melt pools can determine the consolidation extent and possibility of the formation of defects such as porosity and lack of fusion. These defects that arise during printing can adversely affect mechanical properties and surface finish [1]. To ensure high-quality of printed parts and to optimize the process parameters, it is crucial to develop effective approaches for monitoring and quantifying these defects in the multi-track multi-layer LPBF-printing process. In addition, quantifying melt pool geometries enables the correlation of process parameters with printing outcomes, as melt pool dimensions reflect the complex solidification history dictated by processing conditions. Thus, it is highly desirable, for improved process control, to identify and quantify features like melt pools and any defects or anomalies that arise during the AM by analyzing microstructure from cross-sectional samples of the printed components. Examination of the cross-section samples enables detailed observation of the material's internal features and characteristics that may not be evident through surface inspection. Typically, these features are revealed by microscopy techniques such as laser profilometry or optical microscopy. Traditionally, the images generated from these microscopy techniques are analyzed manually leading to a time-consuming process with high cost and subjectivity [3]. One possible solution is to apply image-based quantification approaches to automatically segment and quantify defects and melt pools in the microstructure images of produced parts [4]. Recent advancements in computer vision and machine learning have shown promise for automating image segmentation tasks, reducing human errors, lowering costs, and enhancing efficiency for processing. Deep learning (DL), as a specialized field within machine learning, has become prominent in image processing. While general machine learning algorithms and non-machine learning methodologies struggle to handle image data efficiently. For instance, traditional algorithms used for the identification of microstructural features such as grain boundaries [5, 6] depend on the contrast difference between the grains and boundaries and find it challenging in the case of melt pools segmentation due to the associated poor contrast and noise. On the other hand, DL has its unique capability to capture intricate patterns and complex hierarchical features from raw image data based on the structure and depth of neural networks, without relying solely on the contrast difference of different features. In addition, the deep learning models are highly adaptable to different types of tasks for image processing, such as object detection, classification, and segmentation. These advantages establish DL model as the best suitable method for this research.

Computer vision task aims to interpret and understand the visual information from an image and represent it in a digital format. Several classical computer-vision methods

such as statistics-based visual methods [7–9], gradient-based edge detection algorithms [10–12], and region segmentation using clustering [13] exist. However, these algorithms suffer from limitations of being sensitive to variability in imaging conditions such as lighting, clutter, occlusion [14] and noises, making it challenging to differentiate foreground features from background. This can pose a challenge in AM image processing as well due to the unpredictable and varying quality and noise levels that may occur under different conditions and operations of the collected data. During the big-data era, DL for computer vision, emerges as a sought-after technology in manufacturing. It offers the ability for end-to-end training from raw data, improving the handling of complex segmentation tasks and reducing the reliance on customized processing. [15–20]. Convolutional Neural Network (CNN), as one of the popular model paradigms in DL, has proven to be efficient and effective in learning and extracting local features for processing 2D or 3D images [21]. CNN takes an image as input and perceives it as a collection of pixels arranged in a grid pattern. This model then applies a set of signal processing operations to extract features using convolution in a sliding window manner. The hierarchical structure of CNN allows it to abstract high-level features from input data and train with the gradient-based learning algorithm to optimized weights of the network for better performance. CNN outperforms many state-of-the-art approaches [22, 23] implemented in many general computer-vision tasks, such as image classification [24, 25], object detection [26–29], and segmentation [30, 31].

In AM, CNN-based methods have also been adopted for defect detection and monitoring in various applications. These works can be categorized into three different types of tasks according to the level of difficulty [32]: classification, object detection, and segmentation. Classification problems involve categorizing an image into one or more classes to make decisions. For example, binary classification problem is used to determine whether an image contains defects or not, yielding a true or false answer as a result [33–36]. On other hand, multi-class classification usually requires a model to distinguish between multiple defects. Currently, researchers are attempting to construct DL models to address problems from three different angles: degree of defects, identification of various defect types, and condition-based defects classification [37–39], identification of various defect types [32, 40–43], and condition-based defects classification [44, 45]. Furthermore, the second level of difficulty of the AI-based image processing task is object detection. This task involves not only categorizing the image but also detecting and localizing the objects or defects within it. This type of task usually involves labeling data using bounding box, a rectangular box that are drawn around the region of object. Compared to classification, object detection

requires more complex algorithms to identify and locate the defects accurately [32, 44]. Lastly, segmentation is the most challenging task as it involves recognizing target objects at the pixel level, enabling more fine-grained inspection and monitoring of the AM process from images. Some existing works implemented DL models to detect defects [46], to investigate and predict sub-surface pores [32, 47] and to measure melt pool geometry [48]. Besides, a recent work [49] applied conditional Generative Adversarial Neural Network to quantify structure characterization of melt pools and porosity.

Although most of these works have utilized CNN-based model to investigate the formation of various defects in AM, there are still several limitations. First, many studies examining melt pool characteristics are limited to analyze single-track or single layer samples, which may not capture the complexity of real-world scenarios. Considering that multi-layer multi-track printing involves more physical processes and interactions at the microstructural level, these analyses may not be sufficient. While there are several state-of-the-art DL models available for image segmentation tasks [50–52], previous studies have not explored the comparative performance of applying these different methods to the task of segmentation of the melt pools and defects in laser-based AM processing. Further, these works have not critically evaluated different backbone networks for the efficient utilization of transfer learning capabilities. This is a crucial aspect particularly in case of AM where generation of large data sets for training is impractical. The advantage of transfer learning is that the model trained on the great variation and less biased large training dataset provides a generic representation of learned features and enable faster and efficient learning for new and limited data sets as in the case of AM samples. While some new approaches in DL techniques such as Multi-Fidelity Deep Neural Networks (MF-DNN) [53] potentially can be used for the cases where significant data noise exists or limited data exists, they are apparently more suitable for numerical data and regression problems rather than the image data and vision tasks.

In this work, we aim to address the above gap areas by applying a DL-based framework to automatically detect the melt pool morphology and porosity simultaneously, using the micro-structural images taken from the cross-section of the multi-track multi-layer samples printed by the L-PBF process. This enables us to quantify and thereby gain a better understanding of the impact of both melt pool geometry and porosity on the quality of the printed parts. In addition, seeking consistent-annotation strategy for multi-class segmentation in AM data, multiple labeling strategies have been examined to provide a comprehensive evaluation of the model performance as a function of the annotation strategies. Moreover, various state-of-the-art backbone networks, and DL models were employed to

bring out the comparative evaluation of these networks for the current task. To further examine the robustness of the trained model and identify the minimum amount of data needed to achieve comparable performance, a data-sensitivity test was performed. Overall, this study aims to provide a robust DL-based model and training strategies for improving AM data analysis and quality control for multi-layer multi-track analysis on the L-PBF process.

2 Methods

This section outlines the comprehensive process and methodology employed in the current work for applying the DL-based techniques for the segmentation of the micro-structural images for the identification of melt pool boundaries and pores. The process consists of several key stages, beginning with data acquisition where a range of samples are printed, and image data is collected. The second stage involves data pre-processing steps such as cleaning, annotation, and preparing the training dataset for use to train the model. Following this, the model-building stage involves model training on the prepared data and tuning based on evaluation metrics. Once the model is considered satisfactory, it can be used for inference on unseen data by feeding new inputs to obtain predictions. Finally, post-processing steps are taken on predicted results to allow for statistical analysis based on the automated generated predicted masks. The method flowchart can be seen in Fig. 1.

2.1 Material and specimen preparation

2.1.1 AM experiment

The Trumpf Truprint 1000 machine was used to execute the LPBF fabrication processes (Fig. 2a), which involved using gas-atomized SS 316L powder obtained from CT-PowderRange. The powder's chemical composition is presented Table 1.

The printing was carried out using a bi-directional scanning approach, as shown in Fig. 2b, where the laser beam scan direction was rotated by an angle of 90 degrees for each layer with a fixed laser beam spot size of 55 μm . To ensure the diversity of samples obtained from the builds, a range of processing parameters was selected for the experiments. The selective combination of different processing parameters, including laser power, scanning speed, hatch space, and layer thickness is listed in Table 2.

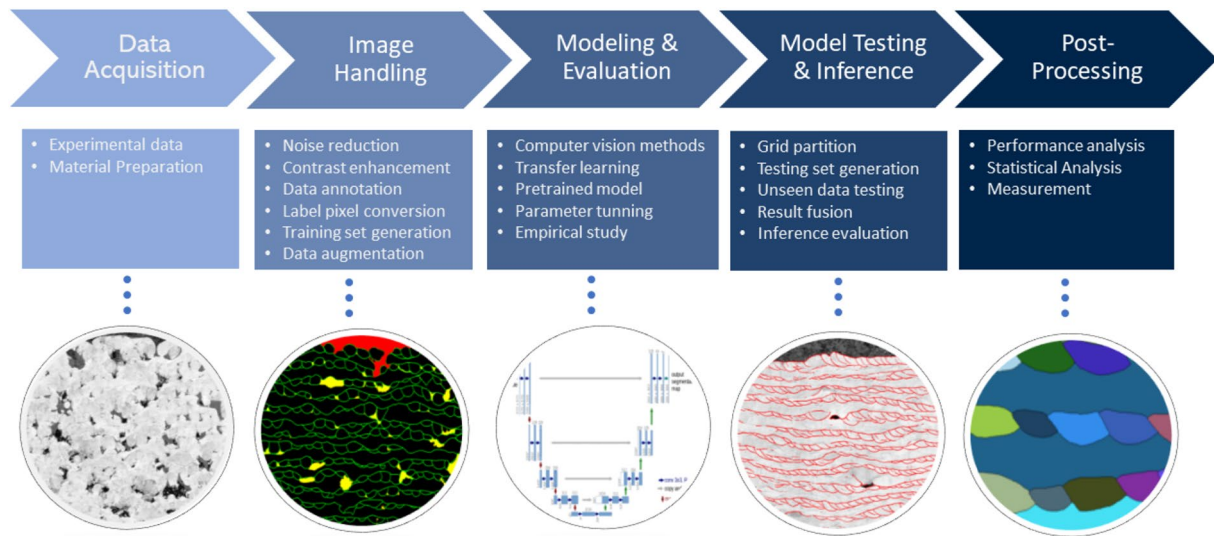


Fig. 1 Flowchart of the proposed scheme of DL-based segmentation of microstructural features (melt pool and porosity) of AM (LPBF) samples

Fig. 2 Schematics of **a** LPBF process **b** Bi-directional scanning used in L-PBF fabrication

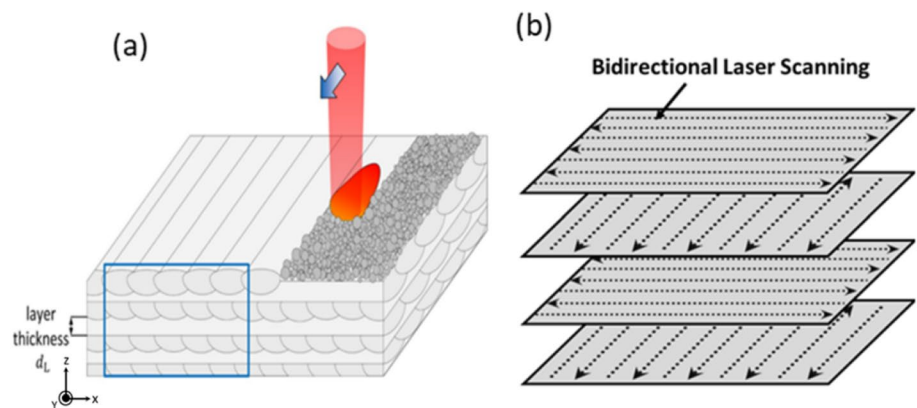


Table 1 Chemical composition of SS 316L powder

Element	Cr	Mn	Mo	Ni	Si	Fe
Content (wt%)	18.0	2.0	2.5	13.0	0.75	Balance

2.1.2 Sample preparation

The sample preparation process involved multiple steps to ensure high-quality samples obtained for subsequent image analysis. First, the printed samples were sectioned along the xz plane using EDM machine (Electrical Discharge Machine). This cutting technique ensures minimal physical distortion and damage to the material while maintaining its microstructure. Next, the sectioned samples were subjected to mechanical polishing using Silicon Carbide (SiC) paper to remove surface irregularities and EDM-induced surface artifacts. This process started with coarse 400 grit SiC paper and progressed to fine 1200 grit SiC paper. Lastly, the final stage involves VibroMet polishing,

where the samples were polished with a colloidal silica solution to enhance the surface quality even further.

2.1.3 Profilometer measurements

Following the completion of standard metallographic polishing techniques on the printed samples, an examination of their surface topographies was carried out using the Keyence VK-X 1000 laser microscope. This state-of-the-art instrument allowed for the capture of high-resolution image data by utilizing a 50× magnification lens which was employed during the examination process.

Table 2 Processing parameters selected in LPBF printing

No	Batch samples	Laser power (W)	Scanning velocity (mm/s)	Hatch space (mm)	Layer thickness (mm)
1	B1C1	100	1000	0.06	0.05
2	B1C2	100	1000	0.1	0.05
3	B1C4	125	1000	0.1	0.05
4	B1C5	125	1000	0.06	0.05
5	B1C6	100	1000	0.08	0.05
6	B1C7	150	1000	0.06	0.05
7	B2C1	150	1000	0.06	0.05
8	B2C2	150	400	0.08	0.05
9	B2C3	150	400	0.06	0.05
10	B2C4	150	600	0.1	0.05
11	B2C5	150	600	0.08	0.05
12	B2C6	150	600	0.06	0.05
13	B2C7	150	800	0.1	0.05
14	B2C8	150	800	0.08	0.05
15	B2C9	150	800	0.06	0.05
16	B2C10	200	1500	0.1	0.05
17	B2C11	200	1500	0.08	0.05
18	B2C12	200	1500	0.06	0.05
19	B3C10	200	2000	0.1	0.03
20	B3C11	200	2000	0.08	0.03
21	B3C12	200	2000	0.06	0.03

2.2 Dataset preparation

To better prepare the data for DL model training and performance analysis, several image processing steps are applied as described below.

2.2.1 Data preprocessing and image enhancement

When training with DL models for computer-vision tasks, image quality plays an important role in the entire process. To better refine the image quality of raw images, image enhancement is an essential step that helps improve the perceptibility of the information in images. [54, 55] has shown that high-contrast images can effectively improve the training of deep neural networks. Accordingly, the operations in this study contain brightness and contrast adjustment for readily accentuating the features of target objects. Besides, noisy data adversely affects the performance of DL models [56]. Hence, Despeckle filter [57] is applied to reduce the visual impact of speckle noise from an image while preserving the details of detected edges. It is worth mentioning that applying filters to reduce noise is always a trade-off between noise reduction and maintaining the complete image information. Thus, this work applied as least smoothing filters as possible to retain the original

information from images maximally and improve perceptual quality to enable efficient labeling and subsequent training.

2.2.2 Data annotation and labeling methods

Image annotation or data labeling involves assigning labels to the objects in images, which could assist the models to train and detect certain patterns based on the visual representation. In current context, porosity requires region-based segmentation whereas melt pool boundary requires localizing class-specific contours. Hence, the pixel-level annotating method was chosen to satisfy these two contrasting requirements. The polygon-annotation method is selected in this work because it is more suitable to meet the requirements by outlining the melt pool border (MP) and filling the enclosed area of porosity [32, 48].

Acknowledging the variability in annotation processes across different scenarios, this study also investigates labeling strategies to identify a standardized method for object annotation in additive manufacturing applications. In AM, there are no consistent rules for labeling, which can make it difficult to evaluate the model performance from heterogeneous data. To better understand how labeling strategies impact model performance and how to improve it, we have attempted five different labeling strategies. The annotation was performed using GIMP [58]. Each target object is individually annotated in separate layers using unique colors (pixel values) for labeling. In the case of multi-class labeling, all the classes are merged together to generate a final mask for each raw image.

Figure 3 shows the labeling examples/strategies. The original input images consist of four objects, including melt pool border (MP), porosities (P), material (MA), and mounting material (MM), an area from the top which does not belong to printed part. MP and P are the two primary targeted objects. MA are the areas inside MP boundaries. Although MM is not a focus of this study, its high similarity in pixel values to porosity may lead to misclassification and potentially affect model accuracy. Based on this assumption, various labeling methods are tested to identify an efficient and optimal approach for annotating the data.

In training a DL model, two types of input are fed into the network. First, the raw images are utilized to preserve their originality and assess the model's performance with such raw heterogeneous inputs. Second, masks are employed to enable the network to learn the features of the target objects. Multiple types of labeling strategies are explored. A variety of samples are used to ensure a more representative and to overcome potential data imbalance issue.

- *MP*: When annotating an image, only the melt pool boundaries are identified and annotated, while the remaining objects are masked out by setting their pixels

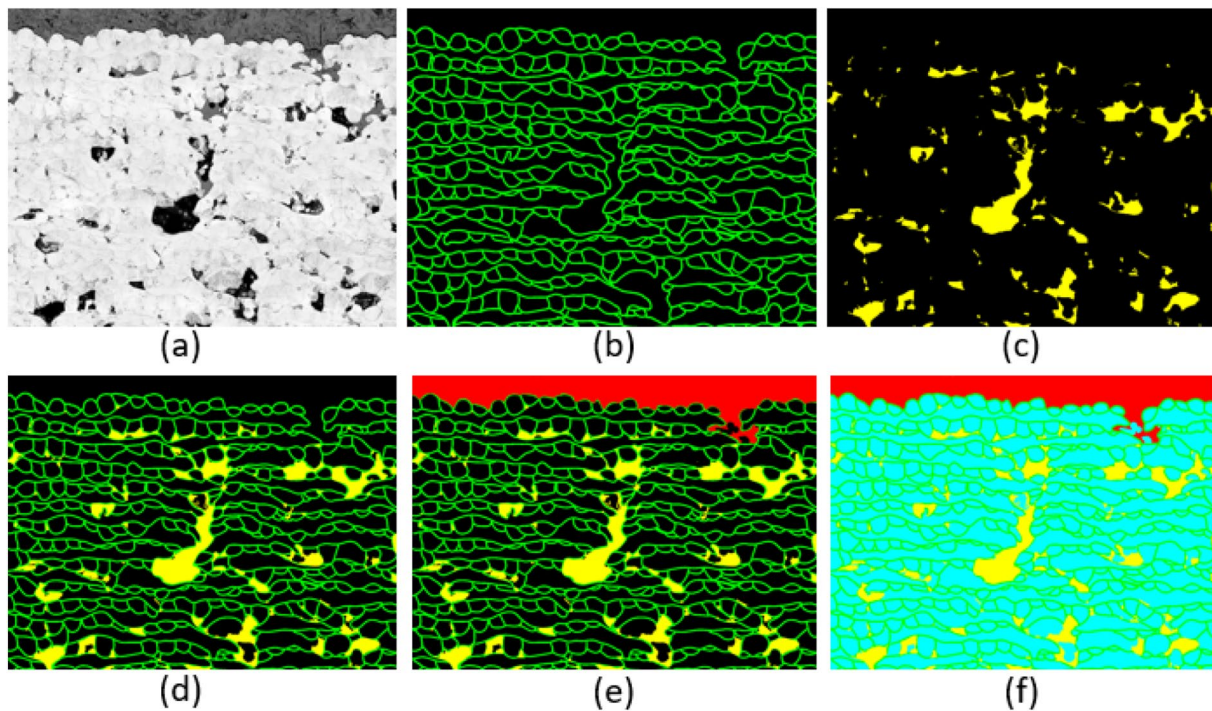


Fig. 3 Example about the labeling in various settings. **a** Example of original raw image **b** The labeling mask only includes the melt pool (indicated by green boundary lines) **c** The labeling mask includes only porosity (indicated by filled yellow area). **d** The labeling mask includes melt pool boundary and porosities. **e** The labeling mask

includes melt pool, porosities, and mounting material (shown in red area). **f** The labeling mask includes all objects in the image: melt pool, porosity, mounting material, and material (shown in light blue color)

to 0, which serve as the background. For this category, the input image and corresponding mask can be seen in Fig. 3a and b, respectively.

- *P*: Similar to the melt pool, *P* represents the porosity class. During annotation, regions of porosity are identified and annotated independently. The raw input image and its corresponding mask are displayed in Fig. 3a and c, respectively.
- *MP + P*: Both melt pool boundary and porosity regions are labeled simultaneously in a single mask as shown in Fig. 3d.
- *MP + P + MM*: Considering the similarity with pixel values of mounting material to other classes, the mounting material are added as a distinct class in addition to the melt pool boundary and porosity on the masks in the hope of distinguishing them to reduce erroneous judgment from raw-image inputs during the training. This mask sample is displayed in Fig. 3e.
- *MP + P + MM + MA*: All the existent objects in the raw images are annotated and formed masks for each pixel segmentation. Figure 3a and f are an example of input image and the corresponding mask.

The mask comprises pixel-wise class labels distinguished by various colors, which are subsequently transformed

into continuous pixel values during processing. The total processed dataset consists of 21 images with corresponding masks. 19 out of 21 images are extracted for the purpose of training and validation of segmentation model and the rest images remain as unseen data and would be used for inference purposes. Each of the raw images has size ranging from 1047×1056 pixels to 2815×2112 pixels.

2.2.3 Generated datasets for model training and inference

The annotated masks are prepared for the training input by assigning unique class numbers (background:0, MP:1, porosity: 2, mounting material: 3 and solidified metal: 4) [59]. The raw images and generated label masks for training purposes are randomly cropped into small patches with sizes of 256×256 pixels. Each of the raw images and corresponding masks are randomly cropped into 100 images. For the inference data, the raw images are cropped contiguously next to each other without overlapping between patches. The masks for inferences are used to evaluate the final performance of predicted results. In order to maintain the spatial dimensions and prevent the loss of important information at the edges of inference image data, padding is added around the edges of the images and corresponding masks [60].

The dataset used for training purposes is partitioned into three subsets: training set, validation set, and testing set. The training set is used to train the model and the validation set can evaluate the model's performance during the training and assist with hyperparameters tuning. The test set is used to evaluate the final performance and generalization ability of the model. In this work, the ratio of dataset division is 80:10:10 with 80% of the data used for training, 10% for validation, and the rest of the 10% for testing.

Furthermore, given the challenge of limited data availability in AM, an extra analysis has been designed to examine sensitivity of the model performance to the quantity of data. Various training data sets are generated by selecting different ratio of data from the available collections Table 3. illustrates the images that were selected for this experiment and the total amount of patches of data used for training models in various settings.

2.3 Deep learning methods

2.3.1 Deep learning-based image segmentation

In this study we employed encoder–decoder type of CNN networks for achieving semantic segmentation considering the specific characteristics of the data being used (microscopy data) and the huge success of these networks in image segmentation tasks of natural images [61].

The first half of the network encodes the input by performing a compression of raw data from high-dimensional image into a low-dimensional representation. And then the decoder tries to decode and up-sample the representation from previously compressed information to construct a segmentation map. The properties of the decoder part naturally fit for image segmentation purposes. One of the advantages of this type of approach is that it can produce sharper boundaries which can delineate objects of different classes more efficiently [62]. This characteristic is particularly suited to our needs for melt pool segmentation and porosity detection, as microscopy data often contains noise and unclear boundaries. Our work utilizes three encoder–decoder networks:

U-Net [50], LinkNet [51], and FPN (Feature Pyramid Network) [52].

U-Net has been employed in a variety of fields such as biomedical image segmentation, satellite imagery or remote sensing [63]. It makes use of a U-Shaped architecture to build a contracting path and a symmetric expansive path. In the contracting process, the network down-samples the input images and extracts growing abstract features with 3×3 convolutions. Then, the network utilizes transposed convolutional layers to up-sample the features to enable precise localization and generate predicted masks with same spatial dimensions as the input images during the second half of the network, where the expanding path is located. The feature maps obtained from the contracting part (down-sampling) are forwarded to the expanding part (up-sampling). This architecture design allows the network to capture both local and global information from images [64].

LinkNet also follows encoder–decoder structure but incorporates several modifications to the network on the basis of U-Net. One of the key modifications that LinkNet makes to the U-Net architecture is the use of residual blocks to combine feature maps from the encoding phase to the corresponding feature maps in the expanding path, instead of merely applying the conventional convolution structure. In addition, LinkNet replaces the concatenation operation used in U-net with an “addition” operation in different layers. The addition operation provides the element-wise addition of the feature maps, which allows the network to focus more on local information. These modifications advance the network to enhance the ability of capturing both coarse and fine-grained features from images and improve the stability and robustness of network.

Feature pyramid network (FPN) is designed for addressing multi-scale problems in image processing tasks. The FPN consists of three parts: a bottom-up pathway, lateral connections, and a top-down pathway. The bottom-up pathway in FPN extracts features from images either from the original input or through a backbone network at a different scale using a feedforward network. Images are progressively down-sampled at each layer to create feature maps while lower layers extract fine-grained details and abstract features can be extracted from higher layers. These feature maps are then processed to construct the feature pyramid in the part of top-down pathway. Next, the top-down pathway progressively up samples the feature maps using transposed convolutions. At each layer, the lateral connection takes feature maps from the bottom-up pathway and combines them with the up-sampled feature maps obtained from the top-down pathway. This combination helps the network to learn information at multiple scales and refine the feature maps to maintain spatial information that might be missing in the down-sampling stage.

Table 3 Data set setting for sensitivity test of data ratio

No. of images	Total cropping patches	Training set	Validation set	Testing set
1	100	80	10	10
3	300	240	30	30
5	500	400	50	50
7	700	560	70	70
9	900	720	90	90
20	2000	1600	400	400

While these three models share some similarities in structure and can be used for segmentation tasks, they also have some differences. For example, compared to U-Net, LinkNet added batch normalization and residual connections in its network architecture to seek for improvement on network performance, but it also contains more parameters than U-Net which may increase the computational cost and memory usage during network training. Compared to U-Net and LinkNet, FPN have more modifications in the network design. It designed to tackle problems on input images with various scales and resolutions. Specifically, to achieve this, FPN is designed to adopt and integrate features from each layer of the backbone network while U-Net and LinkNet only extracts the features from the last layer of encoder. These unique features of different networks make them suitable for different tasks and may result in different effects and performance for AM-specific data. Therefore, training and comparing their performance is indispensable.

2.3.2 Model training optimization techniques

In general, the quality of the ML model performance increases with the quantity and diversity of the input data, as demonstrated by existing deep networks trained on data sets of millions of natural images [61]. However, in the field of manufacturing, collection, and annotation of such huge quantities of data is not practicable leading to low accuracy and high variance in the model performance. To circumvent the deficiencies due to limited quantity of data sets, data augmentation techniques such as random cropping, flipping, random rotation, Gaussian noising, random brightness, and contrast changing, image scaling and shifting were applied.

In model-based optimization, an essential technique, namely transfer learning, employing pre-trained models on large datasets, alongside refining architecture and

hyperparameters tuning [65]. Transfer learning provides a significant advantage by training the model on a diverse and less biased large dataset. This results in a generic representation of learned features, which mitigates overfitting and enhances generalizability [66].

In current work, we have adopted the transfer learning to exploit the benefits of general-purpose feature extractor capabilities of the large pre-trained models. [4, 60]. In this study, ImageNet [61] is utilized for feature extraction. ImageNet is one of the largest and most diverse image datasets, containing over 14 million hand-annotated images. Two state-of-the-art convolutional neural network architectures are implemented in this study, EfficientNet [67] and DenseNet [68]. A summary of these networks is listed in the Table 4. The number that attaches to the network name refers to the depth of layers.

We initiate the training with the weights of these backbone networks (transfer learning) and let training proceed with the AM-specific data set for fine tuning of model weights for the current task using the primary networks stated in Sect. 2.3.1 to learn segmenting the melt pool boundaries and porosities. Other optimization techniques are focused on the model training and hyperparameters tuning. The following table displays the final setting of the training parameters after careful consideration regarding the scale of data and the type of task and multiple rounds of experiments (Table 5).

In this work, the compound loss techniques [49] are applied using the linear combination of Focal loss[69] and dice loss

Table 5 Hyperparameter in model training

Learning rate	Optimizer	Activation function	Batch size	Epochs
1×10^{-4}	Adam	Single class: softmax Multi class: sigmoid	8	50

Table 4 Summary of backbone networks

	Backbone	Description	Strength
DenseNet	DenseNet-201 [68]	A densely connected convolutional networks invent the Dense block where the input of each layer are outputs from all preceding layers. Between each dense block, a transition layer consists of convolution followed by max pooling layer used to change the feature-map sizes	DenseNet try to reduce the impact of information vanishes due to long distance between input and output layer and the design allows to reduce the number of parameters
EfficientNet	EfficientNet b7 [67]	The key design of EfficientNet is the employment of compound scaling which uniformly scales three dimensions (width, depth, and resolution) with a fixed scaling coefficient for a given image. EfficientNet has 8 models from b0-b7. The variant with higher index has more parameters and higher accuracy. “b7” is selected to this study due to our prior experiments	EfficientNet is faster to train and have higher accuracy than other large CNN models in general computer-vision tasks

[70] to evaluate the difference between the prediction and ground truth while aiming for minimizing them during the training.

2.3.3 Model evaluation metrics

To better assess and improve the approach, the evaluation metrics are introduced in the process of the model training and testing. Considering the complexity of the image segmentation task in AM, multiple evaluation metrics are used aiming to provide a comprehensive view for the model's performance.

The evaluation metrics for the proposed semantic segmentation models that adopted in this work are Intersection over Union (IoU) and Dice coefficient (F-Score).

(1) IoU is calculated by the overlapping area between predicted map and ground truth divided by the union of them. For a multi-class segmentation task where multiple objects are being processed simultaneously, the mean IoU of the image is calculated by taking the IoU of each labeled class and averaging them.

$$\text{IoU} = \frac{\sum_{i=1}^N (p_i \times g_i)}{\sum_{i=1}^N (p_i + g_i - p_i \times g_i)} \quad (1)$$

(2) Dice coefficient (F-score) is positively correlated to IoU but IoU tends to penalize single instances of bad segmentation more. The equation of Dice coefficient metrics is as follows:

$$\text{F-score} = \frac{2 \sum_{i=1}^N (p_i \times g_i)}{\sum_{i=1}^N (p_i + g_i)} \quad (2)$$

where p_i and g_i represent the predicted label and ground truth, respectively, for the i -th pixel in the image, where N is the total number of pixels.

3 Results and discussion

3.1 Predictive performance and comparison of various data annotation strategies

3.1.1 Single class segmentation

Initially, we describe the model performances for single class segmentation problem. In this case, the model is trained to segment only one class of the objects (melt pool or porosity). The predictive performances of melt pool-intensive and porosity-intensive tests are displayed in Fig. 4 to show the capability of trained model on the inference data. The predictions for both the melt pool and porosity demonstrate a remarkable ability to match the MP and porosity area of intricate inputs. The results are from a network (U-Net with backbone network EfficientNet b7) trained for 50 epochs. One can notice that the model was able to capture the melt pool boundaries despite relatively poor contrast of the input image, demonstrating the power of this approach in handling realistic and complex input images for the efficient segmentation. The single class accuracies achieved were mean IoU of 68.67% for melt pool and 74.96% for porosity. See Fig. 4. These values are found to be better than the earlier reported IoU values of 0.38 for melt pools [48] where in VGG-19 network was employed for the segmentation task.

While the perceptual similarity between the ground truth and prediction is quite striking, it was observed that

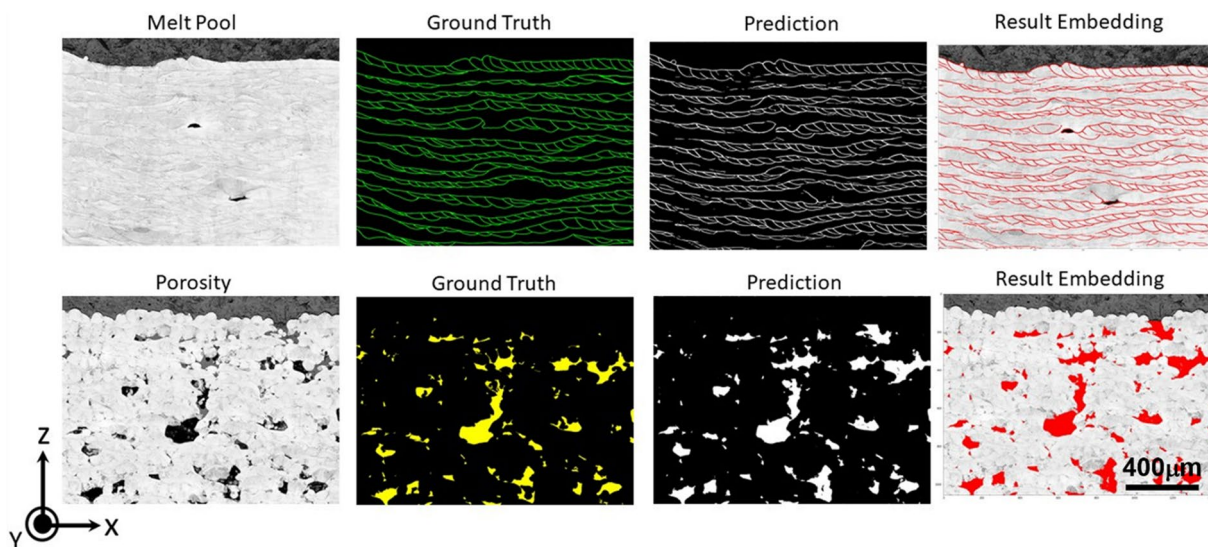


Fig. 4 Predictive results for melt pool and porosity on inference data from best-performed single class model

the predicted melt pool boundaries had some discontinuities at some of the places. This could be attributed to the insufficient contrast of the melt pool boundaries in the input images and also to the imperfections in the manually annotated data. In several instances, manual judgment of the melt pool boundaries was rather difficult and resulted in imperfections in the annotated training data. It may be noted that this is a common issue in manually annotated data sets and may be circumvented by further increasing the data-set size and enhancing the quality of the input images further. In spite of these challenges in the quality and quantity of the input data, the performance of the model in capturing the melt pool boundaries appears to be remarkable and significantly more than any conventional segmentation scheme (using thresholding and morphologic operations) and helps us in

getting statistical distribution of the features of interest to an acceptable accuracy as demonstrated in further sections.

3.1.2 Multi-class segmentation

In multi-class segmentation trials, the combination of MP + P performed poorly compared to other labeling strategies with mean IoU of 59.35% and mean F-score of 65.02%, see Fig. 5. However, after adding more objects as distinct classes, the accuracy was observed to increase as shown in Fig. 5. The hybrid labeling method with MP + P + MM + MA exhibited the highest performance, with a mean IoU score of 72.25% and a mean F-score of 75.79%. The improved performance when incorporating more objects, such as mounting material and material, may be attributed to the potential confusion resolution between foreground and background in AM images, because of availability of higher level of details in the training data. The model can acquire a better understanding of the objects, leading to enhanced segmentation performance. This additional information allows the model to distinguish between target objects and background more accurately and improve its segmentation capabilities.

In summary, single class (Fig. 6b) have more accurate performance while multi-class (Fig. 6a) provides capability of detecting multiple sparse objects simultaneously. This results in tradeoff between accuracy and speed. Due to the nature of multiclass labeling, some areas from multiple objects may overlap with each other causing the pixel missing in one of the objects. This adds to the difficulty in contour detection for melt pool boundaries. This aspect has been demonstrated in Fig. 6, where in inference results on the same test image for the detection of melt pool boundaries was compared for single class vs multi-class segmentation schemes. It is easy to observe that the single class has higher connectivity and continuity in pixels compared to the predicted mask from multi-class segmentation.

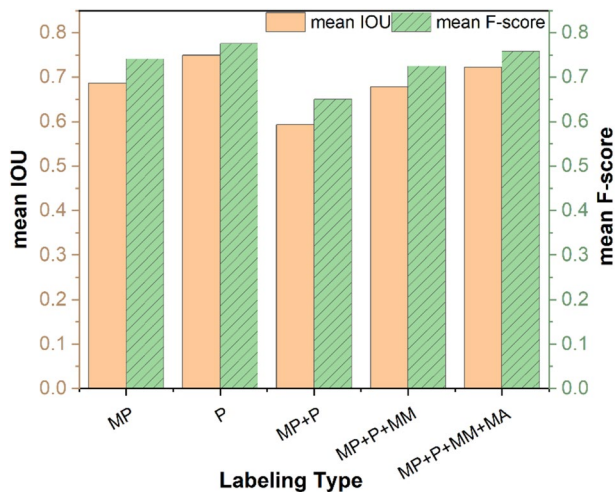


Fig. 5 Performance on inference data with various labeling strategies

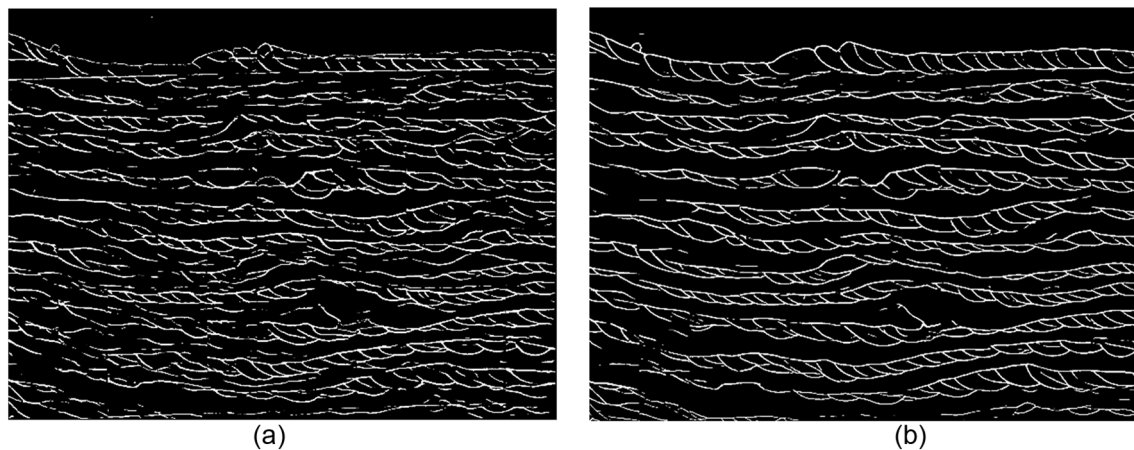


Fig. 6 **a** Melt pool prediction from multi-class segmentation. **b** Melt pool prediction from single class segmentation

As a summary, the suggestions on labeling strategies are as follows. First, independent labeling on interested object is preferred when dealing with complicated and dissimilar objects using CNN. Second, in scenarios where a multi-class segmentation task is desired, it is preferable to incorporate as many distinct classes as possible, particularly when there is confusion between foreground and background pixels.

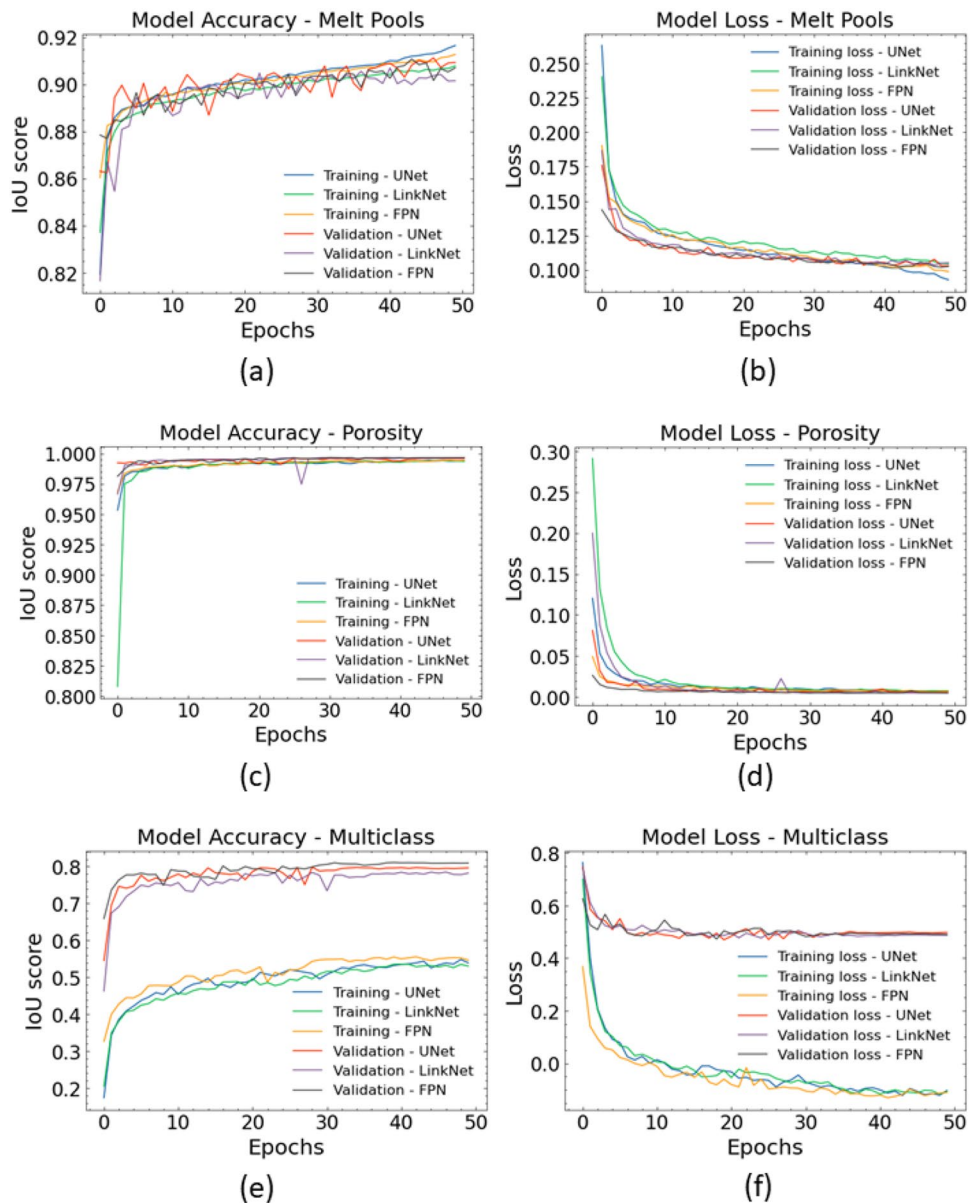
3.2 Insights into model training: training curves and losses

Training curve provides an additional perspective for evaluating the direct process of a model's performance during training and its generalizability. The performances among the three networks are shown in Fig. 7. The figure illustrates

the evolution in accuracy and the training loss over the course of multiple epochs during training and validation stages.

When comparing the single class training curves (Fig. 7a–d) with the multi-class training curves (Fig. 7e–f), it can be observed that for both melt pool and porosity curves, the training and validation curves approach each other closely toward the end of the training process. This convergence implies that the model has achieved optimal performance and is not suffering from significant overfitting or underfitting issues. However, for the multi-class training curves, there remains a gap between the training and validation curves in the results. This indicates that the model may still be overfitting to the training data, resulting in weak generalization when tested on unseen

Fig. 7 Evolution of training and validation scores for various settings (single class and multi-class segmentation) and models (U-Net, LinkNet, FPN). **a, b** validation accuracy and training loss for melt pools (single class segmentation), **c, d** validation accuracy and training loss for porosity (single class segmentation), **e, f** validation accuracy and training loss for multi-class segmentation models



data. Therefore, in order to address this issue and improve the model's generalization performance on unseen data, it may be necessary to acquire and incorporate more data for training the complex multiclass segmentation task. For single class segmentation task, however, current data sets appear satisfactory.

Furthermore, it can be observed that the performance of the segmentation differs for each class within the single class segmentation. The overall accuracy scores for porosity are higher and the loss scores are lower than those for melt pool. This shows that the MP segmentation is a more challenging task in comparison with the porosity. A plausible explanation for this observation is that implemented deep learning algorithms learn based on pixel values, and the porosity area exhibits a notable contrast in pixel values compared to the surrounding region of material unlike the melt pool areas whose contrast is generally weak. Therefore, applying better image-acquisition techniques to increase the melt pool contrast during the data collection (through more optimized etching, illumination, and exposure during image acquisition) and processing stage (digital contrast enhancement using adaptive histogram equalization) may potentially enhance the segmentation accuracy.

In summary, choosing the optimal method to meet the specific requirements often involves a trade-off. Multi-class segmentation allows to segmenting multiple interested objects simultaneously while it also faces some limitation due to the complexity of various characteristics of objects. On the other hand, developing distinct models for individual classes could lead to better accuracy, although it requires multiple rounds of training to achieve results for multiple objects.

3.3 Comparative performance analysis of backbone models

Comparing different combinations of backbone models and network architectures provides insights for a comprehensive evaluation and selection process of the suitable network architecture. In Fig. 8, the combination of three primary network architectures (U-Net, LinkNet, and FPN) and two backbone models (EfficientNet b7 and DenseNet 201) are examined in this study. Regarding the backbone models, the results indicate that the combinations involving EfficientNet b7 generally achieve higher accuracy scores compared to the DenseNet 201 combinations. Specifically, among EfficientNet b7 combinations, both U-Net and FPN show the highest accuracy during inference, while LinkNet exhibits the lowest performance. Similar trend is observed in the DenseNet 201 combinations, with the combination involving U-Net continuing to perform the best. According to the results, it appears that the combination of U-Net's ability to capture fine-grained details and EfficientNet's strong feature

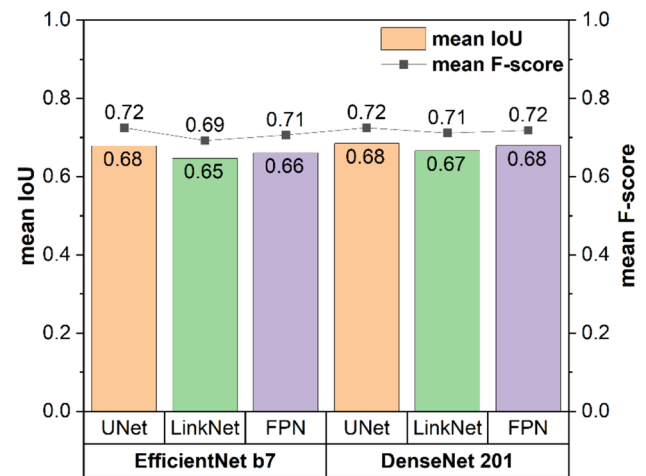


Fig. 8 Performance of different combination of backbone networks on testing data

extraction capabilities maximize the models' performance in learning.

Apart from algorithmic optimization based on the performance accuracy, the additional perspectives to evaluate the capability of a built model are parameters and floating-point operation ratio. Model parameters, in general, are learnt weights during training which keeps track of the size of a model and provide an overview of the scope of model with respect to memory usage in computing. Total parameters refer to the overall number of parameters in the model and trainable parameters represent the number of weights that are being updated during training to optimize the model's performance. The number of trainable parameters can provide insight for the model's capacity to learn, while the total parameters indicate the overall complexity and resource requirements of the model.

The performance of backbone network combination is summarized in Table 6 with reference to their accuracy metrics and total and trainable parameters for a holistic comparison of the different models employed in the current work. The total number of parameters varies across different combinations. The pairing of U-Net and EfficientNet b7 exhibits the highest total parameters, while the combination of LinkNet with DenseNet 201 has the lowest total parameters. When considering trainable parameters, The U-Net and EfficientNet b7 combination has the highest computational cost among the other combinations, having the highest trainable parameters. In addition, comparing different backbone models, it can be observed that the EfficientNet b7 tends to have a larger number of total and trainable parameters compared to DenseNet 201. As expected, the network with higher complexity (U-Net with EfficientNet b7), had given the better performance (in terms of accuracy metrics, see Table 6). These results suggest that the EfficientNet b7,

Table 6 Performance of backbone network selection

Network	Backbone	Testing IoU	Inference IoU	Testing F-score	Inference F-score	Inference MSE	Total parameters	Trainable parameters
U-Net	EfficientNet b7	74.24%	67.85%	80.56%	72.49%	0.0396	75.05 M	74.74 M
	DenseNet 201	73.39%	68.47%	79.76%	72.49%	0.0399	26.38 M	26.15 M
LinkNet	EfficientNet b7	72.41%	64.66%	78.74%	69.23%	0.0392	72.26 M	71.94 M
	DenseNet 201	68.47%	66.60%	74.90%	71.15%	0.0404	22.55 M	22.31 M
FPN	EfficientNet b7	74.24%	66.05%	80.56%	70.62%	0.0403	67.71 M	67.40 M
	DenseNet 201	72.11%	67.91%	78.41%	71.82%	0.0414	21.53 M	21.30 M

Highest values achieved in the respective accuracy metrics are highlighted in bold for readers reference

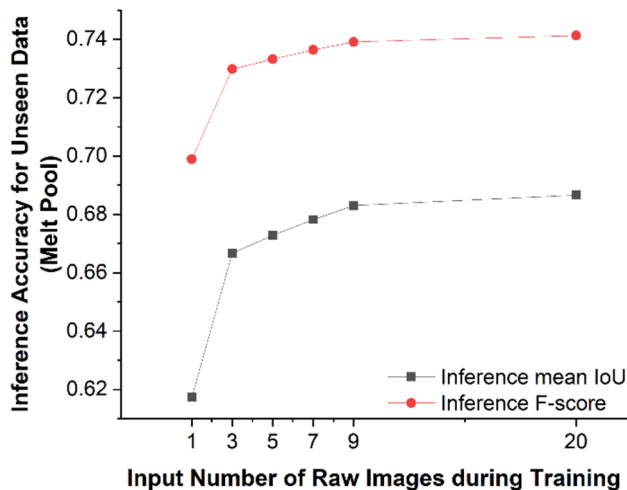


Fig. 9 Performance of Melt pool prediction based on different ratio of training data

with its larger parameter space, may have a higher capacity to capture complex patterns compared to others backbone networks.

3.4 Sensitivity to training data ratio

Figure 9 demonstrates the trend of prediction performance on unseen test images when applying various amounts and ratios of data to train the models. In this case, single class segmentation of melt pools was considered. The results show an increasing accuracy in predictions with an increased number of images in input data. However, the gain in the accuracy saturates beyond 9 images, indicating that there is negligible improvement in the performance even though the input data size is increased to 20. This indicates that mere increasing the data size beyond the optimal value does not proportionally gain in terms of prediction accuracy. Therefore, there exists an optimal data size that provides the best performance for the given task. This finding is valuable as it indicates that collecting a massive amount of data may

not always lead to significant improvements in prediction accuracy.

It is impressive that even with a relatively small data set of nine images, a significant mean IoU of 0.68 was achieved for the complex case of melt pool segmentation. This highlights the effectiveness of the chosen deep learning approach, as it demonstrates the model's ability to learn from limited data and produce accurate predictions. Comparing this result with a recent work that employed a different architecture (U-net with VGG-19), where a mean IoU of 0.63 was achieved, further emphasizes the advancements made in the current study.

3.5 Melt pool quantification from ML-predicted segmentation maps

In an effort to distinguish the form of the segmentation of individual melt pool for further analysis, watershed algorithms are applied to improve the visibility of enclosed melt pool area. Watershed [71] is a classical segmentation algorithm using region-based technique in digital image processing field. It considers an image as a topographic landscape and uses the gray values of their corresponding pixels to decomposes an image into catchment basins. Then each pixel is placed either in a region or a watershed which identified as the boundaries of the object in an image. Watershed algorithm in this work is used for segmenting the melt pool area across multi-layers based on the predicted results of our trained model.

The results of the three trained models, which were examined using the watershed algorithm to segment the melt pool area from the detected MP, are shown in Fig. 10. It can be observed that the performances of melt pool detection with different neural networks are relatively similar among three methods as shown in Fig. 10a–c. The color map (Fig. 10d–f) obtained when watershed algorithm is applied showed minor differences in the final segmented maps. The results from the three models are characteristically similar with very minor and subtle differences. However, among the three networks,

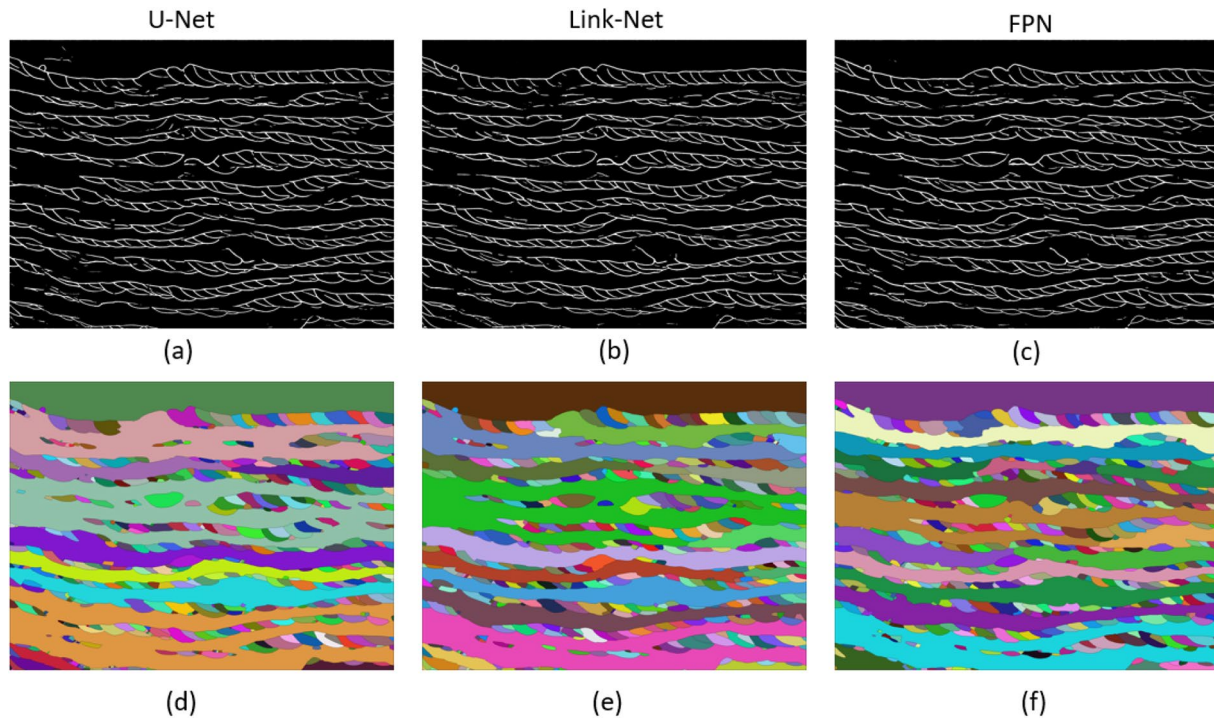


Fig. 10 Colormap of melt pool area segmentation. The top row **a–c** presents the output of the deep learning models for melt pool segmentation **a** U-Net, **b** LinkNet **c** FPN. **d–f** are corresponding unique color maps (using watershed algorithm) of the identified melt pools

segmentation with U-Net captured more details and had better melt pool boundary connectivity.

Overall, the developed models outperform the traditional segmentation algorithms dramatically to segment the regions for melt pools and porosity and makes microscopy image data accessible for automated quantification. Our proposed method has the capability to handle noisy data and produce reliable segmentation results, which is a significant advantage for applications where the quality of AM images is limited. Overall, our study demonstrates a robust and efficient solution for accurate AI-enhanced segmentation of AM microstructure image analysis.

To further evaluate and quantify the effects on model predicted results of melt pool geometry among three networks, Fig. 11 is presented with a comparative analysis of melt pool area distribution, quantified by relative frequency, as displayed by ground truth (Fig. 11a) and three deep learning segmentation models: U-Net (Fig. 11b), LinkNet (Fig. 11c), and FPN (Fig. 11d). The proximity of the distributions generated by the deep learning models to the labeled ground truth indicates a high degree of accuracy in melt pool segmentation. The U-Net model exhibits a highly effective alignment of the proper distribution of melt pool area, reflected in the highest IoU and F-score among the models' testing. The distribution pattern of the LinkNet model is slightly wider than the ground truth but still maintains a similar peak. The FPN model's distribution maintains the

general trend. Overall, the close resemblance across the models, particularly U-Net, to the ground truth highlights the robustness of our proposed methods in dealing with varied forms of defects and complex printing scenarios.

The performance on the distribution of the melt pool area confirms that the predicted results are consistent with the trend of ground truth, indicating the reliability and consistency of our proposed method. To summarize, the obtained segmentation map facilitates the automated quantification of multi-layer, multi-track melt pools from samples fabricated by L-PBF, significantly improving efficiency and minimizing manual intervention. The capability of our methods significantly helps in detecting melt pools on samples that are generated across a range of process parameters involving complicated-solidification processes and various defect scenarios. In addition, current methodology is highly robust and capable of handling noisy data. An important practical implication of this feature is the possibility of applying it in the actual-production settings for potential quality monitoring and performance tuning.

4 Conclusion

This paper presented a deep learning-based approach for defects detection and segmentation for porosity and melt pool geometry from cross-sectional microstructures of

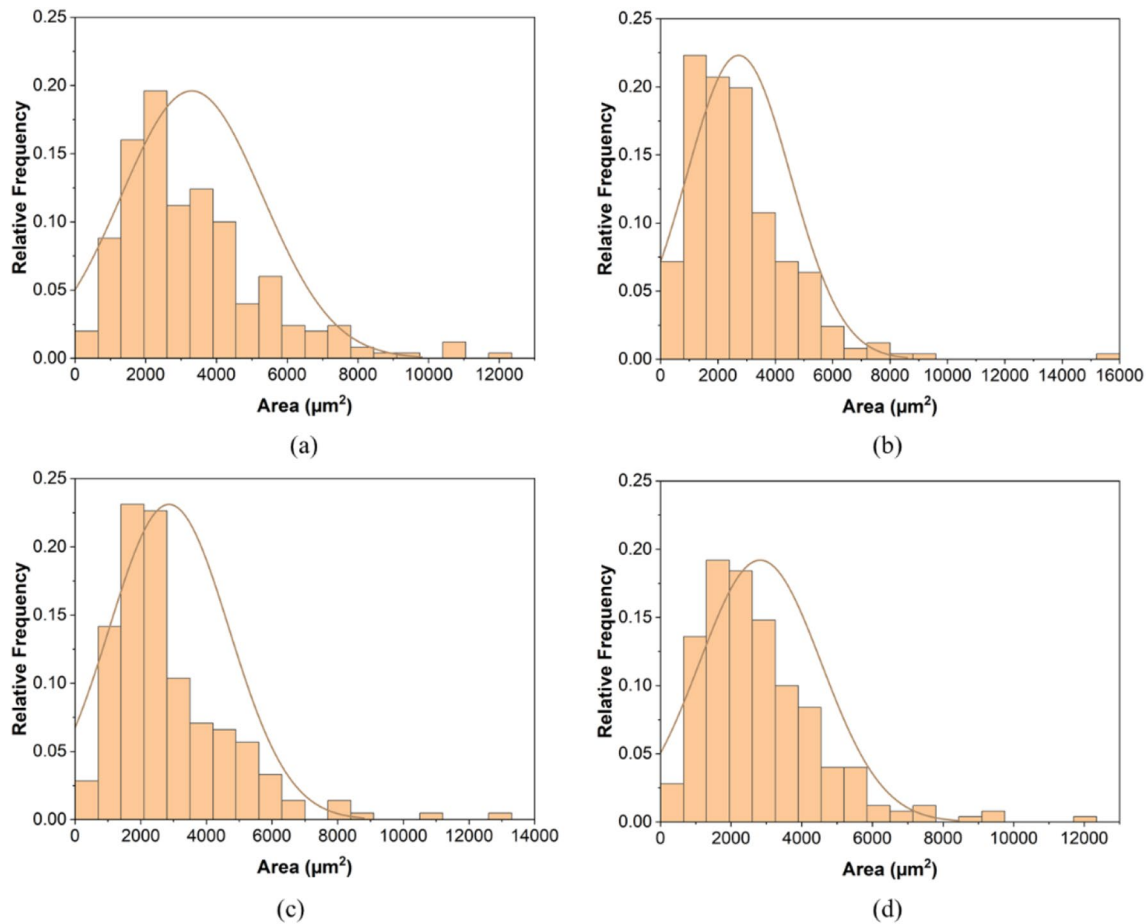


Fig. 11 Area distribution of melt pools. **a** Area distribution from labeled mask (ground truth). **b** Area distribution from prediction with U-net. **c** Area distribution from prediction with LinkNet. **d** Area distribution from prediction with FPN

samples manufactured using laser-powder bed fusion. The proposed method addresses the challenges associated with detecting melt pool boundaries and porosities in noisy microscopy data and segments heterogeneous objects with high accuracy, among the complexities of multi-layer, multi-track microstructures. The method adopts an encoder–decoder based network and utilizes various optimization techniques through a data-centric approach with noise reduction and data augmentation and a model-based optimization that leverages transfer learning with pre-trained models. Our work has made significant progress by enabling the simultaneous detection of multiple features, such as pores and melt pool morphologies. Previous studies [35, 48] have mainly focused on identifying defects in a single layer or segmentation of a single feature in a multilayer setting. In contrast, our approach demonstrates superior capacity to identify multiple objects with significant variations in characteristics, all while maintaining high performance.

The main findings and implications of this work are highlighted as follows:

- The proposed deep learning-based approach presents an opportunity to automate the simultaneous segmentation of melt pools and porosities in 3D printed parts, which could pave the way for effective quality monitoring and quantitative assessment of defects in AM.
- The combination of EfficientNet b7 and U-Net emerged as the most effective choice, among the evaluated deep learning models U-Net, LinkNet, and FPN paired with various backbone networks. This superiority was evidenced by achieving the highest testing IoU score of 74.24%, along with the top testing and inference F-Scores (80.56% and 72.39%, respectively) with limited training data. Availability of large number of parameters in the EfficientNet b7 + U-Net combination was found to be the reason for its observed best performance.
- Our proposed method for melt pool segmentation has demonstrated a significant enhancement over the existing work [48], with our results showing an approximately twofold increase in the IoU score compared to the cited study. Furthermore, our approach to porosity

segmentation is innovative, presenting a novel solution in the absence of comparable existing work.

Acknowledgements Authors acknowledge the infrastructure and support of Center for Agile and Adaptive Additive Manufacturing (CAAAM) funded through State of Texas Appropriation: 190405-105-805008-220.

Data availability The computational and experimental data required to reproduce these findings can be made available upon request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Fu Y, Downey ARJ, Yuan L et al (2022) Machine learning algorithms for defect detection in metal laser-based additive manufacturing: a review. *J Manuf Process* 75:693–710. <https://doi.org/10.1016/j.jmapro.2021.12.061>
2. Liu Z, Zhao D, Wang P et al (2022) Additive manufacturing of metals: Microstructure evolution and multistage control. *J Mater Sci Technol* 100:224–236. <https://doi.org/10.1016/j.jmst.2021.06.011>
3. Holm EA, Cohn R, Gao N et al (2020) Overview: computer vision and machine learning for microstructural characterization and analysis. *Metall Mater Trans A* 51:5985–5999. <https://doi.org/10.1007/s11661-020-06008-4>
4. Qin J, Hu F, Liu Y et al (2022) Research and application of machine learning for additive manufacturing. *Addit Manuf* 52:102691. <https://doi.org/10.1016/j.addma.2022.102691>
5. Anantatamukala A, Mani Krishna KV, Dahotre NB (2023) Generative adversarial networks assisted machine learning based automated quantification of grain size from scanning electron microscope back scatter images. *Mater Charact* 206:113396. <https://doi.org/10.1016/j.matchar.2023.113396>
6. Campbell A, Murray P, Yakushina E, Marshall S, Ion W (2018) New methods for automatic quantification of microstructural features using digital image processing. *Mater Des*. <https://doi.org/10.1016/j.matdes.2017.12.049>
7. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern* 9:62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
8. Sural S, Gang Q, Pramanik S (2002) Segmentation and histogram generation using the HSV color space for image retrieval. In: Sural S, Gang Q, Pramanik S (eds) *Proceedings International Conference on Image Processing Image Processing*. IEEE, p II-589–II-92
9. Rafael C, Gonzalez REW (2018) *Digital image processing*, 4th edn. Pearson
10. Canny J (1986) A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell PAMI* 8:679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
11. Marr D, Hildreth E (1980) Theory of edge detection. *Proc R Soc Lond B Biol Sci* 207:187–217. <https://doi.org/10.1098/rspb.1980.0020>
12. Fukunaga K (1990) *Introduction to statistical pattern recognition*. Elsevier
13. Lloyd S (1982) Least squares quantization in PCM. *IEEE Trans Inf Theory* 28:129–137. <https://doi.org/10.1109/TIT.1982.1056489>
14. Zhang J, Marszałek M, Lazebnik S, Schmid C (2007) Local features and kernels for classification of texture and object categories: a comprehensive study. *Int J Comput Vis* 73:213–238. <https://doi.org/10.1007/s11263-006-9794-4>
15. Wang M, Deng W (2021) Deep face recognition: a survey. *Neurocomputing* 429:215–244. <https://doi.org/10.1016/j.neucom.2020.10.081>
16. Grigorescu S, Trasnea B, Cocias T, Macesanu G (2019) A survey of deep learning techniques for autonomous driving. *J Field Robot*. <https://doi.org/10.1002/rob.21918>
17. Qian R, Lai X, Li X (2022) 3D object detection for autonomous driving: a survey. *Pattern Recognit* 130:108796. <https://doi.org/10.1016/j.patcog.2022.108796>
18. Kumar Y, Koul A, Singla R, Ijaz MF (2022) Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-021-03612-z>
19. Shen D, Wu G, Suk H-I (2017) Deep learning in medical image analysis. *Annu Rev Biomed Eng* 19:221–248. <https://doi.org/10.1146/annurev-bioeng-071516-044442>
20. Yang S, Zhu F, Ling X et al (2021) Intelligent health care: applications of deep learning in computational medicine. *Front Genet*. <https://doi.org/10.3389/fgene.2021.607471>
21. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Van Esesn BC, Awwal AAS, Asari VK (2018) The history began from alexnet: a comprehensive survey on deep learning approaches. [arXiv:1803.01164](https://arxiv.org/abs/1803.01164)
22. Alrfou K, Kordijazi A, Zhao T (2022) Computer vision methods for the microstructural analysis of materials: the state-of-the-art and future perspectives. [arXiv:2208.04149](https://arxiv.org/abs/2208.04149)
23. Dai Z, Liu H, Le QV, Tan M (2021) Coatnet: marrying convolution and attention for all data sizes. *Adv Neural Inf Process Syst* 34:3965–3977
24. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60:84–90. <https://doi.org/10.1145/3065386>
25. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
26. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 779–788
27. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969
28. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes C, Lawrence N, Lee D et al (eds) *Advances in neural information processing systems*. Curran Associates Inc
29. Girshick R (2015) Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, pp 1440–1448
30. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3431–3440
31. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 580–587
32. Wen H, Huang C, Guo S (2021) The application of convolutional neural networks (CNNs) to recognize defects in 3D-printed parts. *Materials* 14:2575. <https://doi.org/10.3390/ma14102575>

33. Zhang B, Jaiswal P, Rai R et al (2019) Convolutional neural network-based inspection of metal additive manufacturing parts. *Rapid Prototyp J* 25:530–540. <https://doi.org/10.1108/RPJ-04-2018-0096>
34. Farhan Khan M, Alam A, Ateeq Siddiqui M et al (2021) Real-time defect detection in 3D printing using machine learning. *Mater Today Proc* 42:521–528. <https://doi.org/10.1016/j.matpr.2020.10.482>
35. Westphal E, Seitz H (2021) A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. *Addit Manuf* 41:101965. <https://doi.org/10.1016/j.addma.2021.101965>
36. Snow Z, Diehl B, Reutzel EW, Nassar A (2021) Toward in-situ flaw detection in laser powder bed fusion additive manufacturing through layerwise imagery and machine learning. *J Manuf Syst* 59:12–26. <https://doi.org/10.1016/j.jmsy.2021.01.008>
37. Li J, Zhou Q, Huang X et al (2023) In situ quality inspection with layer-wise visual images based on deep transfer learning during selective laser melting. *J Intell Manuf* 34:853–867. <https://doi.org/10.1007/s10845-021-01829-5>
38. Zhang B, Liu S, Shin YC (2019) In-process monitoring of porosity during laser additive manufacturing process. *Addit Manuf* 28:497–505. <https://doi.org/10.1016/j.addma.2019.05.030>
39. Li X, Jia X, Yang Q, Lee J (2020) Quality analysis in metal additive manufacturing with deep learning. *J Intell Manuf* 31:2003–2017. <https://doi.org/10.1007/s10845-020-01549-2>
40. Scime L, Beuth J (2018) A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Addit Manuf* 24:273–286. <https://doi.org/10.1016/j.addma.2018.09.034>
41. Angelone R, Caggiano A, Teti R et al (2020) Bio-intelligent selective laser melting system based on convolutional neural networks for in-process fault identification. *Procedia CIRP* 88:612–617. <https://doi.org/10.1016/j.procir.2020.05.107>
42. Baumgartl H, Tomas J, Buettner R, Merkel M (2020) A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring. *Prog Addit Manuf* 5:277–285. <https://doi.org/10.1007/s40964-019-00108-3>
43. Cui W, Zhang Y, Zhang X et al (2020) Metal additive manufacturing parts inspection using convolutional neural network. *Appl Sci* 10:545. <https://doi.org/10.3390/app10020545>
44. Zhang Y, Hong GS, Ye D et al (2018) Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring. *Mater Des* 156:458–469. <https://doi.org/10.1016/j.matdes.2018.07.002>
45. Caggiano A, Zhang J, Alfieri V et al (2019) Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Ann* 68:451–454. <https://doi.org/10.1016/j.cirp.2019.03.021>
46. Davtalab O, Kazemian A, Yuan X, Khoshnevis B (2022) Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection. *J Intell Manuf* 33:771–784. <https://doi.org/10.1007/s10845-020-01684-w>
47. Ertay DS, Kamyab S, Vlasea M et al (2021) Toward sub-surface pore prediction capabilities for laser powder bed fusion using data science. *J Manuf Sci Eng Trans ASME*. <https://doi.org/10.1115/1.4050461>
48. Schmid S, Krabusch J, Schromm T et al (2021) A new approach for automated measuring of the melt pool geometry in laser-powder bed fusion. *Prog Addit Manuf* 6:269–279. <https://doi.org/10.1007/s40964-021-00173-7>
49. Peles A, Paquit VC, Dehoff RR (2023) Deep-learning quantitative structural characterization in additive manufacturing. *arXiv:2302.06389*
50. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18. Springer International Publishing, pp 234–241
51. Chaurasia A, Culurciello E (2017) LinkNet: Exploiting encoder representations for efficient semantic segmentation. *IEEE Vis Commun Image Process*. <https://doi.org/10.1109/VCIP.2017.8305148>
52. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2117–2125
53. Li Z, Montomoli F (2023) Surrogate modeling and uncertainty quantification based on multi-fidelity deep neural network. *Reliab Eng Syst Saf*. <https://doi.org/10.1016/j.ress.2024.109975>
54. Cowley B, Pillow JW (2020) High-contrast “gaudy” images improve the training of deep neural network models of visual cortex. *Adv Neural Inf Process Syst* 33:21591–21603
55. Maini R, Aggarwal H (2010) A comprehensive review of image enhancement techniques. *arXiv:1003.4053*
56. Liang W, Tadesse GA, Ho D et al (2022) Advances, challenges and opportunities in creating data for trustworthy AI. *Nat Mach Intell* 4:669–677. <https://doi.org/10.1038/s42256-022-00516-1>
57. Schneider CA, Rasband WS, Eliceiri KW (2012) NIH image to ImageJ: 25 years of image analysis. *Nat Methods* 9:671–675. <https://doi.org/10.1038/nmeth.2089>
58. GIMP - GNU image manipulation program. <https://www.gimp.org/>. Accessed 29 Mar 2023
59. Takahashi R, Matsubara T, Uehara K (2020) Data augmentation using random image cropping and patching for deep CNNs. *IEEE Trans Circuits Syst Video Technol* 30:2917–2931. <https://doi.org/10.1109/TCSVT.2019.2935128>
60. Chollet F (2017) *Deep learning with Python*. Manning Publications, New York
61. Deng J, Dong W, Socher R et al. (2009) ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp 248–255
62. Ghosh S, Das N, Das I, Maulik U (2020) Understanding deep learning techniques for image segmentation. *ACM Comput Surv* 52:1–35. <https://doi.org/10.1145/3329784>
63. Yan C, Fan X, Fan J, Wang N (2022) Improved U-net remote sensing classification algorithm based on multi-feature fusion perception. *Remote Sens (Basel)* 14:1118. <https://doi.org/10.3390/rs14051118>
64. Wang Z, Zou N, Shen D, Ji S (2020) Non-local U-nets for biomedical image segmentation. *Proc AAAI Conf Artif Intell* 34:6315–6322. <https://doi.org/10.1609/aaai.v34i04.6100>
65. Tan C, Sun F, Kong T et al (2018) *A survey on deep transfer learning*. Springer International Publishing, Cham
66. Goodfellow I, Yosuga Bengio AC (2016) *Deep learning*. The MIT Press, Cambridge
67. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International conference on machine learning*. PMLR, pp 6105–6114
68. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4700–4708
69. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*, pp 2980–2988
70. Sudre CH, Li W, Vercauteren T et al (2017) Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Lecture Notes Comput Sci*. https://doi.org/10.1007/978-3-319-67558-9_28

71. Beucher S (1992) The watershed transformation applied to image segmentation. *Scan Microsc* 1992:28

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.