

## Vivian Wen Hui Wong<sup>1</sup>

Engineering Informatics Group,  
Department of Civil and Environmental  
Engineering,  
Stanford University,  
Stanford, CA 94305  
e-mail: vwwong3@stanford.edu

## Max Ferguson

Engineering Informatics Group,  
Department of Civil and Environmental  
Engineering,  
Stanford University,  
Stanford, CA 94305  
e-mail: maxferg@stanford.edu

## Kincho H. Law

Engineering Informatics Group,  
Department of Civil and Environmental  
Engineering,  
Stanford University,  
Stanford, CA 94305  
e-mail: law@stanford.edu

## Yung-Tsun Tina Lee

Systems Integration Division,  
National Institute of Standards and Technology  
(NIST),  
Gaithersburg, MD 20899  
e-mail: yung-tsun.lee@nist.gov

## Paul Witherell

Systems Integration Division,  
National Institute of Standards and Technology  
(NIST),  
Gaithersburg, MD 20899  
e-mail: paul.witherell@nist.gov

# Segmentation of Additive Manufacturing Defects Using U-Net

*Additive manufacturing (AM) provides design flexibility and allows rapid fabrications of parts with complex geometries. The presence of internal defects, however, can lead to the deficit performance of the fabricated part. X-ray computed tomography (XCT) is a nondestructive inspection technique often used for AM parts. Although defects within AM specimens can be identified and segmented by manually thresholding the XCT images, the process can be tedious and inefficient, and the segmentation results can be ambiguous. The variation in the shapes and appearances of defects also poses difficulty in accurately segmenting defects. This article describes an automatic defect segmentation method using U-Net-based deep convolutional neural network (CNN) architectures. Several models of U-Net variants are trained and validated on an AM XCT image dataset containing pores and cracks, achieving a best mean intersection over union (IOU) value of 0.993. The performance of various U-Net models is compared and analyzed. Specific to AM porosity segmentation with XCT images, several techniques in data augmentation and model development are introduced. This article demonstrates that U-Net can be effectively applied for automatic segmentation of AM porosity from XCT images with high accuracy. The method can potentially help improve the quality control of AM parts in an industry setting. [DOI: 10.1115/1.4053078]*

*Keywords: smart manufacturing, defect detection, additive manufacturing, convolutional neural networks, X-ray computed tomography (XCT) images, machine learning, artificial intelligence, data-driven engineering, machine learning for engineering applications*

## 1 Introduction

With years of development, additive manufacturing (AM), also known as 3D (three-dimensional) printing, has become an important technology in the manufacturing industry. The layer-by-layer process provides design flexibility and allows the manufacturing of parts with complex geometries [1,2] and multifunctional properties [3]. The fabrication process, however, comes with the possibility of internal defects, which are often difficult to detect. Layer-wise quality control is therefore very important for AM, as the internal defects could lead to undesirable properties in the fabricated part, resulting in deficit performance [4]. An automated procedure that can identify defects of parts fabricated using AM is essential.

Current trends with nondestructive inspection (NDI) approaches often involve process monitoring through the installation of a large array of sensors and then analyzing and detecting failures using the collected sensor data [5–7]. These in situ methods require the analyses of multiple signal types, and their correlations to final part quality are not yet well understood. Alternatively, ex situ NDI techniques, such as X-ray computed tomography (XCT), are used to evaluate a completed build and offer a more reliable

characterization of the AM part. XCT has emerged as perhaps the preferred technique for measuring the properties of a completed AM build. It can be used to visualize internal structures and identify small pores and flaws in an AM part [8]. Obtaining useful images and segmentation labels from XCT scans, however, involves manual thresholding, making the process unscalable to a large number of samples.

Although many conventional methods to identify small defects remain difficult to implement in a manufacturing setting, the segmentation of defects in XCT images can be automated using computer vision and deep learning techniques. Here, the segmentation of AM defects refers to the ability to characterize and differentiate between porosity-indicative volumes and a fully dense part. Effective segmentation of defects enables more efficient identification, labeling, and sorting of such volumes. Defect segmentation can be framed as an image segmentation problem, which assigns each 2D pixel or 3D voxel of an image to a class. For defect segmentation, each pixel or voxel can be classified as either the fully dense background or porosity using a deep learning model.

Convolutional neural networks (CNNs) have been commonly used for segmentation problems [9] and have been shown effective in many domains, including everyday objects [10], satellite imagery [11], surface crack detection [12], structural stresses [13], manufacturing defects in casting [14], and melt pools from welding [15]. Most of these segmentation problems deal with 2D image data, but the biomedical domain, with its need to segment volumetric images such as computed tomography (CT) and magnetic resonance imaging (MRI) scans, poses the need for segmenting 3D

<sup>1</sup>Corresponding author.

Contributed by the Computers and Information Division of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received July 30, 2021; final manuscript received November 16, 2021; published online December 10, 2021. Assoc. Editor: Mahesh Mani.

This work is in part a work of the U.S. Government. ASME disclaims all interest in the U.S. Government's contributions.

images [16]. 3D CNNs have demonstrated potential in volumetric medical image segmentation [16–18]. Among existing 3D CNN methods, those with an U-shape architecture, also known as U-Net variants, have achieved excellent performance in several medical image segmentation tasks. The combination of encoder–decoder architecture and skip connections allow U-Net to fuse both high-resolution and low-resolution feature maps for learning, which is very suitable for tasks with a relatively small number of training samples [19]. U-Net variants have been used for images from CT and MRI scans to X-ray, ultrasound, and microscope images and have been successfully applied in a wide variety of medical image segmentation applications, such as brain tumor segmentation [20] and 3D chest CT image segmentation for COVID-19 screening [21]. Over the past 4 years from 2017 to 2020, the popularity of U-Net has grown over 20 folds in publications [22]. Images taken from metal AM parts, similar to their medical imaging counterparts, are volumetric and have comparable levels of contrast. Therefore, 3D CNNs that do well on medical image segmentation could possibly benefit AM image segmentation as well.

Despite the similarities between AM and medical imaging, AM presents many unique challenges. AM defects are pores and faults that are usually small (relative to the volumetric size of the part) and have highly irregular geometries. Furthermore, the sparsity of defects varies significantly among samples. In addition, because of the cost and manual effort needed to produce labeled AM datasets, very few AM datasets are publicly available. The lack of large dataset poses a huge challenge to the adoption of machine learning approaches, which typically require a large number of training data to converge on a reasonable model. Other challenges, such as background noise, low porosity level, and image resolutions, may require special attention to data preprocessing and model developments. Despite these challenges, the mainstream adoption of machine learning methods for defect detection of AM parts is essential to the development of fast and reliable quality control procedures.

Motivated by the need for a defect segmentation method for quality control and inspired by the success of 3D CNNs in medical image segmentation, we applied the 3D U-Net model with existing defect labels to automatically segment defects in XCT images of unknown AM samples [23]. In this article, we focus on AM defects such as pores and cracks, or any internal, possibly defect-indicative, volumetric voids in the part. We show that the 2D U-Net model, trained using 2D planar images, performs well and achieves high accuracy in terms of the mean intersection over union (IOU) measure. Our results demonstrate that both 2D U-Net and residual 3D U-Net can reach high accuracy of 0.993 on the validation set. However, 2D U-Net may be better for some applications as it is easier to train and faster to evaluate. The contribution of this study is therefore not only to propose a method to automatically segment AM porosity with high accuracy but also introduce techniques to augment the 3D U-Net models that can be used to directly perform porosity segmentation of a 3D volumetric part, which is particularly useful for AM parts that have complex geometries.

The content of this article is based on a paper presented at the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC-CIE),

2021 [24]. The rest of this article is organized as follows: Sec. 2 provides an overview of related works. Section 3 describes the AM defect dataset that is used in this study. Section 4 describes background information on CNNs and the U-Net architecture, as well as the results in applying the U-Net models. Section 5 presents the approach taken to improve the performance of 3D U-Net models, including data augmentation and model development techniques. Finally, this article is concluded with a brief summary and discussion in Sec. 6.

## 2 Related Works

With processing, 3D images can be sliced into 2D and vice versa, thereby allowing 2D CNNs to segment volumetric images [16]. The most commonly used CNN architectures for 2D segmentation problem are region-based and fully convolutional network based (FCN based) [9]. Region-based CNN (R-CNN), such as the Mask R-CNN, is an example of the former [9,10]. Since regions need to first be extracted, described, and then classified, these methods are generally more computationally expensive [25]. Conversely, FCN-based methods directly learn a mapping from input to output pixels, without proposing regions [26]. U-Net is a CNN model that extends the FCN architecture, achieving excellent performance, for example, in the segmentation of the ventral nerve cord [27].

Despite the success of 2D CNN models, it has been suggested that since many medical images are 3D in nature, slicing them into 2D images before training loses information on the correlation between slices [28]. To that end, 3D FCN-based segmentation architectures, such as 3D U-Net [17] and V-Net [16], that train on volumetric medical images have been developed. While 3D CNN models can leverage information between slices, several disadvantages exist in comparison to 2D CNNs. 3D CNNs lack pretrained models, leading to less stable training [28]. The patch-wise predictions in 3D are also more time consuming to generate, compared to predictions in 2D. Furthermore, it has been pointed out that 2D U-Net may outperform 3D U-Net when the data are anisotropic [29]. To that end, in this study, we compare the performance of 2D U-Net and 3D U-Net models on the same AM defect dataset.

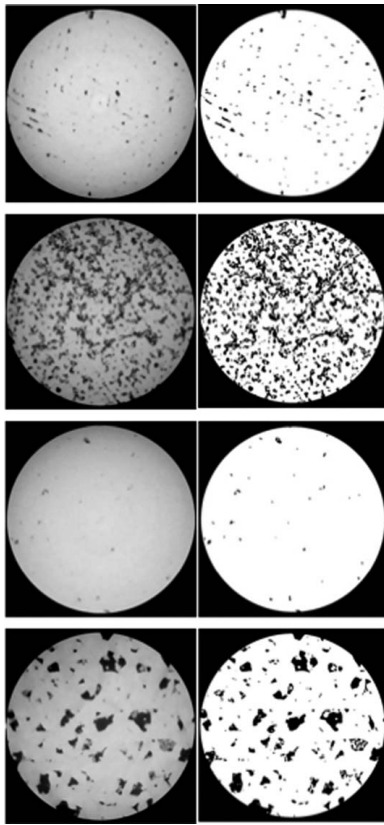
A few related works have been done to automatically detect AM porosity using CNN. 2D detection and classification on porosity using slices of camera images have been reported [30]. CNN has also been used to analyze acoustic emissions during AM processes [31]. More recently, 2D U-Net has been deployed to an AM XCT dataset similar to that used in this study and achieved a mean IOU of 0.92 [32]. In this article, we not only present the novel CNN techniques for porosity segmentation but also present their applicability for achieving accurate results in the domain of AM XCT image segmentation, particularly, for 3D volumetric parts.

## 3 The Dataset

The AM defect dataset used in this study was introduced by Kim et al. [33] and is publicly available [34]. The dataset consists of XCT images of four cobalt-chrome alloy, cylindrical AM specimens, created in a laboratory setting to investigate pore structures. Table 1 details the image size and porosity of each specimen. The metallic cylinders were produced using laser-based powder bed fusion. Artificial pores and cracks were produced by changing AM scan speed and hatch spacing. By changing the process parameters, specimens were processed to have varying porosity. Then, XCT images of the specimens are taken. Each specimen's set of images consists of 8-bit grayscale images of 2D slices of XCT imagery. These images are 16-bit raw images obtained using XCT reconstruction processed by adding a  $3 \times 3 \times 3$  median 3D filter and a nonlocal means filter [35,36]. To obtain ground truth labeling of the defects, Bernsen local thresholding [37] was used to process the 8-bit images. The local contrast threshold parameters of the thresholding process are computed by relating average noise

**Table 1** Details of the AM defect datasets

Specimen	Distance between 2D slices (pixel)	Image volume after 3D reconstruction ( $D \times W \times H$ ) (pixel)	Porosity (%)
Sample 1	0.00245	$900 \times 980 \times 1010$	1.00
Sample 2	0.00277	$900 \times 988 \times 1013$	19.03
Sample 3	0.00243	$900 \times 984 \times 1010$	0.42
Sample 4	0.00252	$749 \times 984 \times 1010$	10.90



**Fig. 1** Examples of images from AM defect datasets: left, processed XCT images; right, segmentation masks are on the right

value to local contrast threshold as explained by Kim et al. [33]. Figure 1 shows examples of the images and the corresponding labels.

The purpose of obtaining the XCT images and thresholding for the labeling of defects is to use them as inputs and ground truth reference for CNN models. CNN models use the images as inputs and produce predicted segmentation masks, classifying defect and

background pixels or voxels, and compare the prediction results with the ground truth to obtain a loss function, which is then minimized through an iterative training process.

To evaluate the applicability of 3D CNN model for volumetric images, the 2D XCT images are concatenated into a 3D volumetric image, restoring the original cylindrical form of an AM sample, as shown in Fig. 2. The AM defect dataset has the following characteristics to be noted:

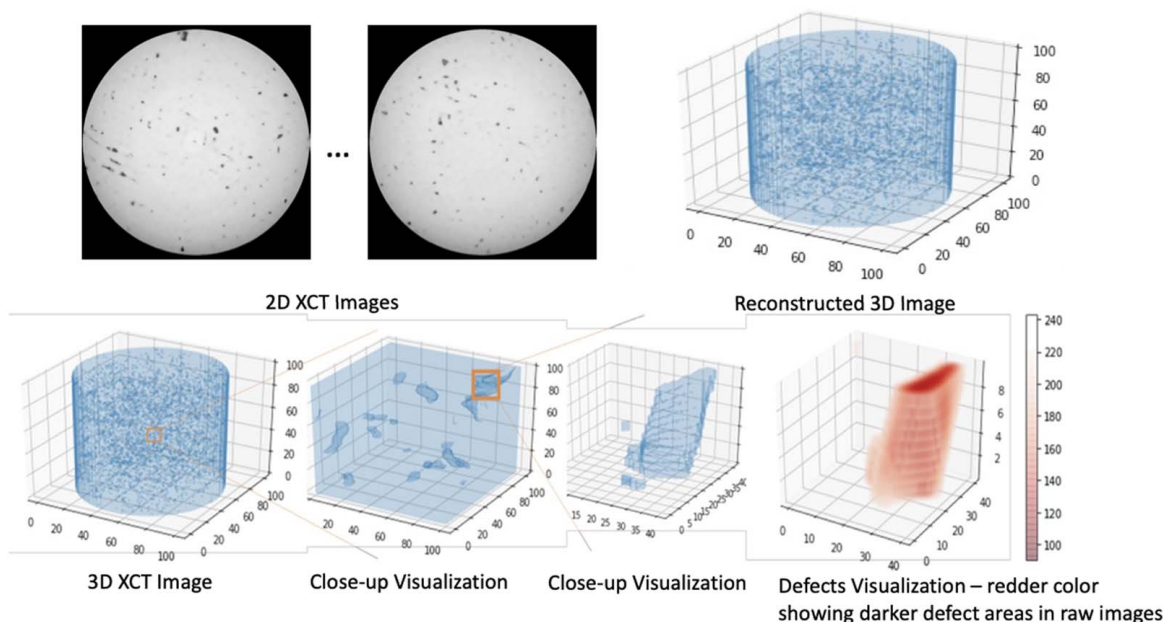
- The standard deviations of the pores on the  $z$ -axis of some samples are not the same as those of the  $x$  and  $y$  axes, meaning that the shape and the distribution of pores may be anisotropic [33].
- The 3D structure of the defects gives limited information about the location according to the ground truth labels, as the labels are generated by thresholding 2D images.
- As shown in Fig. 1, the geometries of the defects can look highly irregular.
- Percentage of porosity indicates that there is an imbalance in the number of porosity and background voxels.

Altogether four specimens with 2D XCT images are available for the study. Three specimens (samples 2–4) are used for training and one specimen (sample 1) is used for validation of the trained models. Since the samples range vastly in porosities, the validation sample is selected because its percentage of porosity is neither the minimum nor the maximum.

#### 4 Convolutional Neural Networks and U-NET

Developments in CNNs in the past decade have significantly improved the ability to perform image classification, detection, and segmentation in many domains. This section first gives a brief overview of deep CNNs. We then introduce the U-Net architecture, which is the architecture that inspires many of recent domain-specific works on image segmentation [22].

**4.1 Convolutional Neural Networks.** A CNN is a type of deep neural network that consists of several layers, where each layer uses mathematical operations, such as convolution, to convert the input to a feature map. CNNs have been commonly employed and operated on 2D images and have recently been



**Fig. 2** Reconstruction of a 3D AM image from 2D XCT images

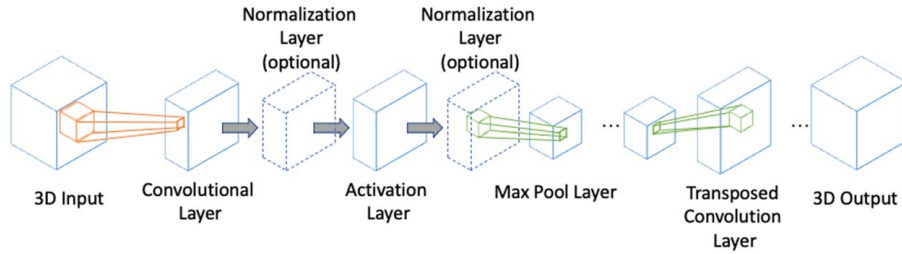


Fig. 3 Example of a 3D CNN built by combining various types of layers

extended to the study of 3D images. The idea of training a 2D or a 3D CNN model is identical, but with the following distinctions:

- (1) The convolving kernels in 3D CNNs are 3D with width, height, and depth ( $W \times H \times D$ ), whereas the kernels in 2D CNNs are 2D with width and height ( $W \times H$ ) only.
- (2) When convolving, a 3D kernel moves in three directions, along all three axes of the input image and its feature maps. A 2D kernel moves in two directions, along the axes corresponding to  $W$  and  $H$  dimensions.

Figure 3 shows an example a 3D CNN architecture with multiple types of layers. A layer  $l$  of a neural network can be written as a function parameterized by parameters  $\theta^{(l)}$ :

$$h^{(l)} = f^{(l)}(h^{(l-1)}; \theta^{(l)}) \quad (1)$$

where  $h^{(l)}$  is the output feature map of layer  $l$  and  $h^{(0)}$  is an image tensor. A layer can be a convolutional layer, which uses a parameterized kernel to convolve with the layer's input. In a convolution layer, the dot product of the kernel and the input at each spatial location is taken. The stacked layers approximate a complex function of the image input and the output  $h^{(L)}$  at the final layer  $L$ , representing the model's prediction. With the stacking of multiple layers, a parameterized function mapping the input image to the prediction can be created. To introduce nonlinearity in the function approximator, a nonlinear activation function is commonly applied to the output of a layer:

$$h^{(l)} = \sigma(f^{(l)}(h^{(l-1)}; \theta^{(l)})) \quad (2)$$

where  $\sigma$  is a nonlinear function. Common choices for  $\sigma$  are the sigmoid function and the rectified linear unit (ReLU) function.

Another type of layer is a normalization layer, such as batch normalization (BN) or group normalization (GN) layer. A normalization layer normalizes its input to improve the speed and stability of training neural networks [38]. BN performs the normalization along the batch and spatial locations. On the other hand, GN, which is more robust than BN, divides the input's channels into groups and performs normalization for each group. GN alleviates the limitation of BN that smaller batch size leads to larger errors [39]. Instance normalization (IN) is another technique that normalizes across spatial locations [40].

Max pool and transposed convolution layers can also be used in CNN models to, respectively, downsample or upsample the input tensor spatially. A max pool kernel draws the maximum at each of the input's spatial locations. A transposed convolution multiplies the input at each spatial location with the kernel and adds the result to the layer's output at the same location.

A deep neural network is trained by minimizing a loss function. The loss function measures the amount to which the prediction differs from the ground truth. During training, the model's parameters,  $\theta = \{\theta^1, \dots, \theta^L\}$ , are updated using the gradient of the loss function, which, thereby, must be differentiable. This method of calculating gradients with respect to the parameters is generally known as backpropagation [41].

**4.2 2D and 3D U-Net Models.** Due to its excellent performance, U-Net is a popular CNN architecture not only in the medical [20–22,27] but also in nonmedical domains such as satellite imagery [42]. The design of U-Net is characterized by two properties: The U-shaped structure formed by an encoder and a decoder network, as well as the skip connections that connect the corresponding encoders and decoders. Figure 4 shows the overall structure of a U-Net architecture. Implementation in this study closely follows this general structure, but with changes in design details of the encoders, decoders, upsampling, and downsampling.

The U-Net architecture consists of two main parts. The contracting path on the left consists of encoder modules and downsampling operations that increase the number of feature maps produced as the number of layers increases. The expansive path on the right consists of decoder modules and upsampling operations that decrease the number of feature maps as the number of layers increases. Although other variations (such as normalization) exist, the encoder and decoder modules are normally convolutions with activation functions. The encoder and decoder modules that have the same resolution are connected with a skip connection, combining their outputs to produce the input for the next decoder module.

The U-Net architecture's modular design allows for flexibility in altering its modules. The classical 2D U-Net's encoder and decoder modules are double convolutions using 2D kernels and a ReLU activation [27]. Conversely, 3D U-Net, described in Ref. [17], deploys

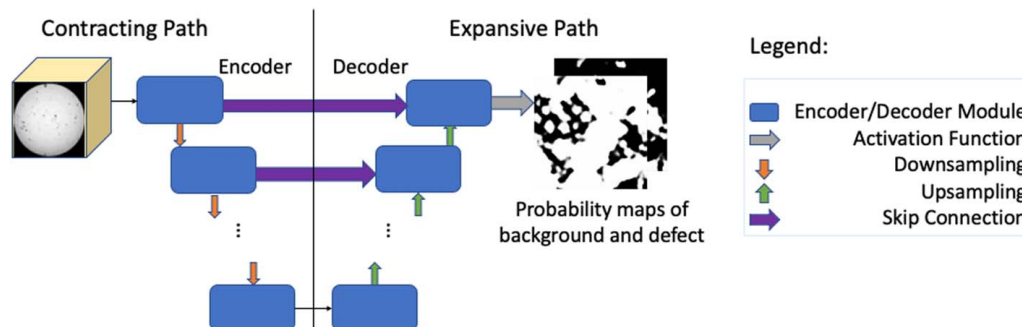


Fig. 4 Schematic of a general U-Net architecture

**Table 2 Details of U-Net base configuration**

Model	Encoder/decoder	Downsampling	Upsampling
2D U-Net	$(3 \times 3 \text{ convolution} + \text{BN} + \text{ReLU}) \times 2$	$2 \times 2$ max pool	Bilinear with a scale of 2
Vanilla 3D U-Net	$(3 \times 3 \times 3 \text{ convolution} + \text{BN} + \text{ReLU}) \times 2$	$2 \times 2 \times 2$ max pool	$2 \times 2 \times 2$ transposed convolution
3D U-Net with GN	$(3 \times 3 \times 3 \text{ convolution} + \text{ReLU} + \text{GN}) \times 2$		
Residual 3D U-Net	$(3 \times 3 \times 3 \text{ convolution} + \text{GN} + \text{ReLU}) \times 3$		

3D convolutions and adds batch normalization to its encoder and decoder modules. Furthermore, the ResUNet-a presented in Ref. [43] discovered that residual connections could improve the performance of U-Net and can help reduce the vanishing gradient problem [44]. Therefore, in ResUNet-a, residual connections were added to the encoder and decoder modules of the U-Net architecture. Combining residual connections and 3D U-Net has been shown to perform well in several medical imagery segmentation tasks [18,44].

**4.3 Implementation.** To assess the performance of 2D and 3D U-Net on AM porosity segmentation, we train and evaluate several U-Net configurations, namely, the 2D U-Net, vanilla 3D U-Net, 3D U-Net with GN (instead of BN in the vanilla configuration), and 3D U-Net with residual connections, as summarized in Table 2. The training and inference of the models are conducted on NVIDIA Tesla T4 GPUs with 100 GB RAM and 4 Intel virtual CPU on Google Cloud Platform.

The 2D U-Net follows the same implementation as described in Ref. [27]. The model's encoder and decoder modules are each a double convolution: twice stacking a 2D convolutional layer, followed by a ReLU activation.  $3 \times 3$  kernels are employed for the convolutional layers, and  $2 \times 2$  max pooling is used for downsampling. Some minor modifications from the standard implementation are made: First, to save memory consumption, bilinear upsamplings are used instead of transposed convolution. Furthermore, the 2D convolutions are zero padded with a one-pixel border to preserve the features of the edges. Finally, a BN layer is added after each 2D convolutional layer and before the ReLU to improve stability. Inputs to the 2D U-Net are the raw 2D images and masks as given in the dataset. The training minimizes cross-entropy loss with an RMSprop optimizer [45], and a learning rate of 0.0001 is used. The 2D U-Net implementation is adapted from a publicly available PyTorch implementation [46].

The vanilla 3D U-Net, on the other hand, follows the implementation by Cicek et al. [17]. The model follows a similar architecture as the 2D U-Net, but with 3D convolutions. The double convolutional layers consist of a 3D convolutional layer, followed by a BN layer and a ReLU nonlinearity layer, all stacked twice to form the double convolution. The second 3D U-Net model is a variant that uses GN as the normalization layer and places the GN layer after the ReLU activation layer. The third model, residual symmetric 3D U-Net, follows the implementation by Lee et al. [18], which introduces residual skip connections in the modules and modifies the upsampling and downsampling techniques. Inputs to the 3D U-Net models are 3D images, constructed by stacking the 2D slices as described in Sec. 3. Due to memory constraint, each training sample is a  $128 \times 128 \times 128$  patch randomly sampled from the 3D image. Stride sizes are  $32 \times 32 \times 32$  to overlap the patches and ensure that information is not lost. The input patches are normalized, randomly flipped, and rotated before training. Network outputs and targets are compared using the cross-entropy (CE) loss. Each model is trained with an initial learning rate of 0.0002 that decays at a rate of a half at the 600th, 1000th, and 1400th iterations. The networks are trained via the Adam optimizer [47]. A weight decay factor of 0.0001 is used. The batch size and the group size are set to one for BN and GN layers, respectively. All modifications to the models are conducted using a publicly available implementation of the 3D U-Net architecture [48].

**4.4 Experimental Results.** The prediction accuracy of each of the aforementioned models is evaluated using the mean IOU metric, comparing the accuracy of a predicted segmentation with the ground truth or labeled mask. Table 3 presents the mean IOU and the training time to achieve the accuracy for the AM datasets. The 2D U-Net outperforms the 3D U-Net models, which could be due to the fact that our dataset is anisotropic and thus, as suggested in Ref. [26], favors the performance of 2D U-Net. Among the 3D models, the residual 3D U-Net model requires the longest training time but performs slightly better than the other 3D U-Net models.

We observe several limitations posed by the 3D models. Figure 5 shows an example patch sampled by the residual 3D U-Net from the validation data. It can be seen that due to the large size of the images, the  $128 \times 128 \times 128$  patches only capture the shape of defects partially. This could limit the model's ability to predict based on the defect's relative spatial location. Sharp edges of the irregular defects are often misclassified, which could be an indication that the additional axis of information that 3D CNNs leverage does not compensate for the loss of global information in the  $W-H$  plane. We also provide a 2D image segmented by the 2D U-Net in Fig. 5, showing that the 2D U-Net's predictions are mostly accurate.

Some drawbacks can also be observed on both the 2D and 3D models. The challenges observed from the predictions using the AM defect dataset are as follows:

- (1) *Variation in sizes:* AM defects can range from hardly visible, very small voids to large voids, as shown in Fig. 1. Small defects are challenging for segmentation using CNN because there are inherently less voxels of these smaller defects for training, and they are difficult to distinguish from background noises.
- (2) *Lack of training voxels:* Following the previous point, there are much fewer defect voxels than the background voxels, which can cause complications. The background voxels located outside the rim of the cylinder are considered trivial. However, because they are naturally dark, they are always thresholded as defects. As shown in Table 1, the specimen with the highest porosity has less than 20% porosity, meaning that there is a significant class imbalance in the training examples.
- (3) *Highly irregular geometry:* It can be viewed in Fig. 1 that the shapes of defects are highly irregular, often consisting of sharp edges and light color rims. This poses difficulties for a CNN model to infer the correct geometry from surrounding voxels, and the boundaries of such irregular shapes are difficult to identify.

**Table 3 Validation mean IOU and average training time per epoch on the base U-Net models**

Model	Training time (h)	Validation mean IOU
2D U-Net	<b>0.70</b>	<b>0.993</b>
Vanilla 3D U-Net	6.58	0.863
3D U-Net with GN	14.00	0.881
Residual 3D U-Net	19.97	0.884

Note: Bold values show that the 2D U-Net outperforms the 3D U-Net models as dataset is anisotropic.

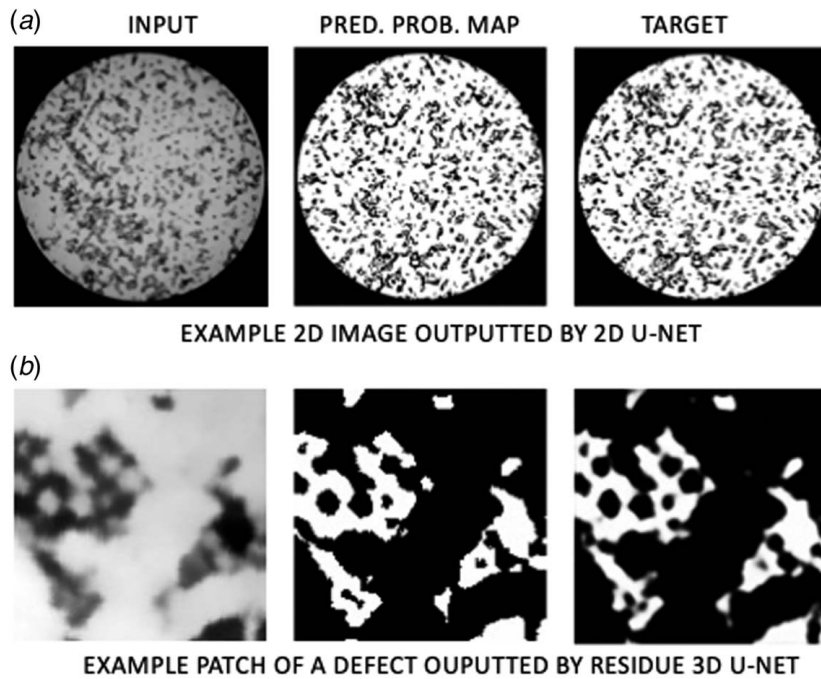


Fig. 5 Examples of segmentation results outputted by (a) 2D U-Net and (b) residual 3D U-Net

(4) *High resolution*: The resolution size of the input images in the referenced AM dataset is very large. Table 1 presents the number of voxels and the array shape of each specimen image. This not only leads to high memory consumption during preprocessing and training but also discards the possibility to conduct downsampling before training, as downsampling would lose valuable information on the already rare small defects.

In Sec. 5, we propose a number of enhancements on the dataset to improve the performance of the U-Net models.

## 5 Data Augmentation and Model Developments

Although the results in Table 3 show that 2D U-Net outperforms 3D U-Net on the dataset used in this study. A 3D U-Net model could be useful to directly make predictions on AM volumetric parts with complex geometries. This section describes an approach that can enhance the performance of 3D U-Net models on segmenting AM defects based on the nnU-Net, which is a framework that performs preprocessing, U-Net configuration, training, and postprocessing for image segmentation [49]. Figure 6 shows how choices

of data enhancement techniques can help address the four identified challenges of AM defect segmentation. We also perform the same techniques on the 2D U-Net to observe their effects. The five techniques as shown in Fig. 6 are as follows:

- *Nonzero cropping*: The images are cropped into regions where voxel values are nonzero. Caused by the high image resolution, each AM specimen image consists of a large number of voxels. Such large arrays of voxels induce heavy computational cost. Cropping out nonzero voxels at borders can reduce the size of arrays while keeping the data containing valuable information for training. However, since convolution is done on rectangular images, some voxels outside of the cylinders are preserved after cropping.
- *Dice + CE loss*: A loss function that sums the Dice loss and CE loss is used [50]. Dice loss is a commonly used loss function for segmentation. Here, the objective function is set as the Dice score, an evaluation metric for accuracy that considers class imbalance. However, since training is done in patches, we cannot calculate the Dice score of the entire image based on any single patch. An estimated Dice score, derived from combined patches, could be an inaccurate estimate of the

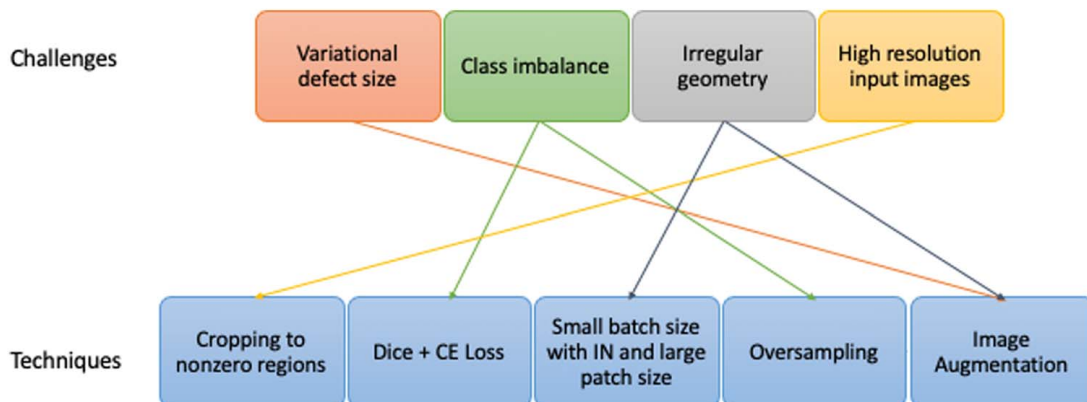


Fig. 6 Techniques used to address segmentation challenges posed by AM defects

**Table 4 Details of enhanced U-Net configuration**

Model	Encoder/decoder	Downsampling	Upsampling
2D U-Net (patched)	$(3 \times 3, \text{stride } 2 \text{ convolution} + \text{IN} + \text{leaky ReLU}) \times 2$	Done through strided convolution	$2 \times 2$ transposed convolution
3D U-Net	$(3 \times 3 \times 3, \text{stride } 2 \text{ convolution} + \text{IN} + \text{leaky ReLU}) \times 2$		$2 \times 2 \times 2$ transposed convolution
Residual 3D U-Net	$(3 \times 3 \times 3, \text{stride } 2 \text{ convolution} + \text{IN} + \text{leaky ReLU}) \times 2$		

true Dice score and lead to unstable training. Conversely, CE loss, another commonly used loss function, measures the degree at which the prediction differs from the true label. It is found empirically that combining CE loss and Dice loss improves segmentation quality [51]. Therefore, to address both training stability and the imbalanced number of defects and background voxels, the loss function is selected to be the sum of the Dice loss and the CE loss functions as follows:

$$L_{dc} = 1 - \frac{\sum_{i \in I} h_i y_i}{\sum_{i \in I} h_i + \sum_{i \in I} y_i} \quad (3)$$

$$L_{ce} = -(y \log(h) + (1 - y) \log(1 - h)) \quad (4)$$

$$L_{total} = L_{dc} + L_{ce} \quad (5)$$

where  $L_{dc}$  is the Dice loss averaged over all batches,  $L_{ce}$  is the cross-entropy loss,  $L_{total}$  is the total loss,  $h$  is the model prediction, and  $y$  is the ground truth.

- **Batch and patch size:** Classification of voxels at boundaries of irregular defects is a difficult task for the network. Typically, a larger training patch size means that more contextual information from surrounding voxels is incorporated when computing the weights. Capturing the full shape of a defect could also lead to less confusion over the boundaries. A larger patch size is therefore desirable in training, but results in a reduction of the batch size. Hence, the batch size is selected to be small. While batch normalization [38] is often used in CNN training to improve robustness and convergence, but, because of the small batch size, instance normalization [40] is used. Finally, the size of kernels is calculated by limiting the total size of feature maps to the limited GPU memory budget.
- **Oversampling:** Since we have less voxels of defects than voxels of background, the (class) imbalance in training data leads to the lack of training data, thereby impacting the accuracy of segmentation. Oversampling resolves the issue by sampling examples containing the rarer class, in this case, the defects, more often than the more dominant class. When sampling patches for training, we ensure that the rarer class is sampled more frequently: Patches are sampled such that at least one of the patches or one-third of the patches in a batch, whichever is greater, is guaranteed to contain a randomly selected defect voxel, with the rest of the batch being randomly sampled.
- **Image augmentation:** Multiple data augmentation techniques are used during training. Images used as inputs to the network are normalized to ensure that each voxel has a similar distribution. This adds robustness and improves convergence during training. In our approach, each image is normalized by subtracting the mean and dividing by the standard deviation of voxels in that image. Images are randomly rotated and scaled. Furthermore, we add additional noise into the dataset to improve robustness. With a certain probability controlled by a random number generator, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, Gamma augmentation, and mirroring are applied. These augmentation techniques help robustness of the network to generalize defects with varying defect geometries, such as aspect ratios, colors, and shapes. Details and specifics of these augmentation techniques have been described by Isensee et al. on nnU-Net [49].

In addition to the data enhancements made earlier, several minor modifications to the original U-Net architecture are made. The

ReLU activation functions are replaced with Leaky ReLU, and downsampling is implemented as strided convolution. Deep supervision is used in training, which adds an additional term for loss in some larger feature maps of the decoder. These modifications are useful design choices to facilitate training.

In the last step of the approach, we train a CNN model that utilizes the five data enhancement techniques and the architectural design choices mentioned previously. To train a model, we sample minibatches and train iteratively to optimize the layer parameters over the Dice + CE loss function.

### 5.1 Implementation

The models implemented with data enhancement techniques are presented in Table 4. All models are trained end-to-end and without pretraining, with weights initialized using the initialization procedure described by He et al. [52]. Stochastic gradient descent with Nesterov momentum [53] is used to optimize the learning. The initial learning rate is selected at 0.01 and decays throughout training at a rate of  $9 \times 10^{-6}$  per epoch. Each epoch is defined as 250 training iterations on the minibatches. The total number of epochs is determined based on the convergence of losses. The training loss is calculated by summing cross-entropy loss and batch Dice loss. Since trade-off exists between runtime and loss reduction, training is terminated at 56 epochs, when all models have reached a plateau in losses.

Given the goal of a small batch size, and constrained by the limited GPU's capacity employed in this work, the 3D U-Net and the Residual 3D U-Net use a batch size of 2, with each patch size being  $128 \times 128 \times 128$ . The 2D U-Net uses a batch size of 3, with the patch size of  $1024 \times 1024$ . Input images are augmented using the previously mentioned image augmentation techniques conducted on the fly during the training process. The inference procedure is patch based and uses the same patch size used during training.

### 5.2 Experimental Results

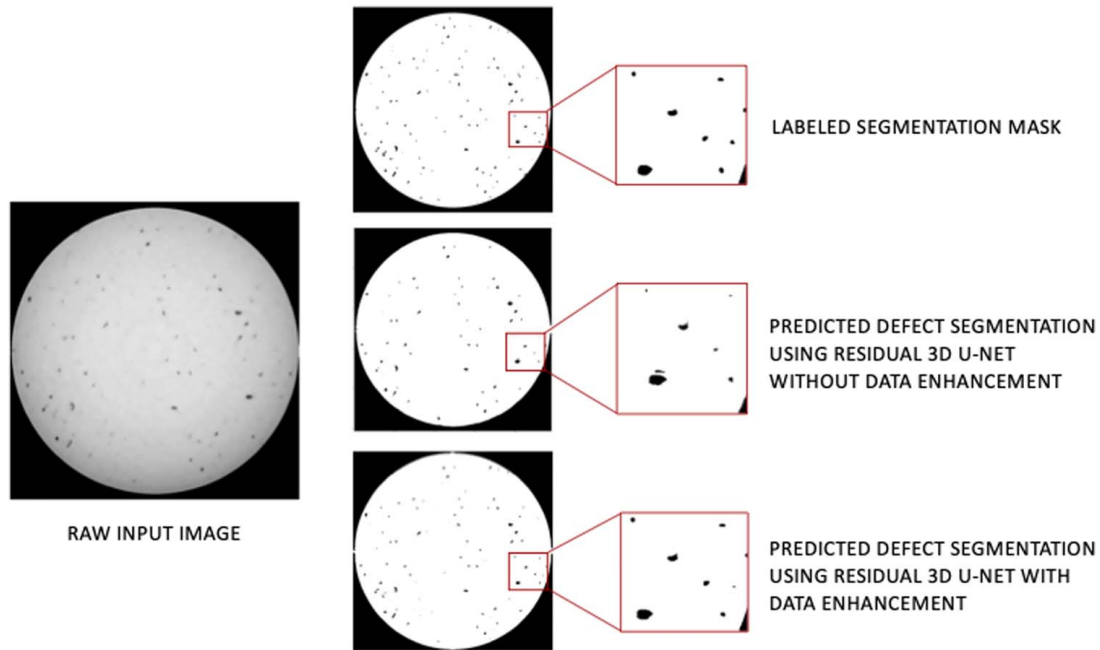
The performances of the three enhanced U-Net models are presented in Table 5 where the mean IOU evaluation scores and the amount of time taken for training are reported. As shown in the table, the residual 3D U-Net model, with a mean IOU of 0.993, achieves the highest accuracy, and is comparable to the nonpatched 2D U-Net as presented in Table 3. The training of the 3D models requires, as expected, much more time than the 2D counterpart, which, with patch-based sampling, also takes longer time than the original 2D U-Net model.

Figure 7 shows a slice of the segmentation mask outputted by the Residual 3D U-Net model. It can be observed that most defects have been segmented by the model. The prediction resembles well with

**Table 5 Validation mean IOU and average training time per epoch on enhanced U-Net models**

Model	Training time (h)	Validation mean IOU
2D U-Net (patched)	4.55	0.988
3D U-Net	21.06	0.992
Residual 3D U-Net	20.61	<b>0.993</b>

Note: Bold value shows that the residual 3D U-Net model, with a mean IOU of 0.993, achieves the highest accuracy.



**Fig. 7 2D slice of a sample segmented by residual 3D U-Net with and without data enhancement**

the labeled mask and is able to segment the complex geometries of most identified defects. However, it should be noticed that defects with very light colors in the input have more ambiguous labels, and therefore, those voxels may not necessarily be classified correctly by the model.

## 6 Summary and Discussion

This article presents the use of U-Net models for automatic detection of AM defects using XCT images. Given the small number of specimens with varying levels of porosities, the U-Net architecture with a combination of encoder, decoder, and skip connectors that allow the fusion of high- and low-resolution feature maps appears to perform very well even with a very modest GPU for model training and inference. Using the dataset available in this study, the 2D U-Net achieves accurate segmentation with the shortest training time. Although the 2D U-Net seems to be the best fit for this AM defect dataset, one must note that the dataset contains several characteristics as described in Sec. 3, which could lead to the 2D U-Net outperforming the 3D U-Net models. 3D U-Net models may become more effective with other datasets and scenarios, for example, when the geometry of the AM fabricated part is complex or when the CT images are much noisier. In practice, AM parts have more complex geometry than the cylindrical specimens employed in this study. 3D model would allow better differentiation between intended and unintended porosity, for example, for parts that have internal features such as holes and channels inside.

Based on the dataset employed in this study, a number of challenges for defect segmentation in AM are identified. The challenges include the large variation in defect sizes, very small percentage of porosities in the specimens, highly irregular shapes and geometries of defects, and high memory consumption due to image resolutions. With minor modifications in network architectures, the mean IOU increased substantially for the 3D U-Net models. We attribute the improved accuracy to the various data enhancement techniques used, including additional preprocessing, oversampling, image augmentation, as well as the change in the design of the loss function. These techniques are purposely tailored toward improving prediction given the domain-specific challenges. We argue that these techniques take on a more significant role than minor changes in network architecture design when deploying 3D U-Net models on the same dataset and suggest that these techniques should be considered in future

related works to improve model performance. Furthermore, for situations that deem necessary, attention modules can be introduced to potentially further improve model accuracy [54].

In summary, while conventional manual or thresholding methods for AM defect segmentation remain tedious and unscalable, this article has presented a method for automatic volumetric segmentation of AM specimens—a challenging task given: the complex geometries of the specimens, the poor contrast and lighting resulting from measuring metal specimens, and the imbalance of defect and background classes associated with the images. A high predictive accuracy with a mean IOU of 0.993 is achieved by the 2D U-Net model and the residual 3D U-Net model with data enhancements. The high accuracy of the method demonstrates the potential of deep learning models to be applied to aid the quality control of AM parts in practice.

## Acknowledgment

This research is partially supported by the Measurement Science for Additive Manufacturing Program at the National Institute of Standards and Technology (NIST), US Department of Commerce, Grant Number 70NANB19H097 awarded to Stanford University. The authors would like to thank Dr. Felix Kim of NIST for providing the 3D printing data sets used for the experimental study. Certain commercial systems are identified in this article. Such identification does not imply recommendation or endorsement by NIST nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting U.S. Government or corporate organizations.

## Conflict of Interest

There are no conflicts of interest.

## Data Availability Statement

The data and information that support the findings of this article are freely available at: <https://doi.org/10.18434/M32162>.

## References

- [1] Gibson, I., Rosen, D., and Stucker, B., 2010, *Additive Manufacturing Technologies*, Springer, New York.
- [2] Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T. Q., and Hui, D., 2018, "Additive Manufacturing 3D Printing: A Review of Materials, Methods, Applications and Challenges," *Composites, Part B*, **143**, pp. 172–196.
- [3] Leung, Y., Kwok, T., Li, X., Yang, Y., Wang, C. C. L., and Chen, Y., June 2019, "Challenges and Status on Design and Computation for Emerging Additive Manufacturing Technologies," *ASME J. Comput. Inf. Sci. Eng.*, **19**(2), p. 021013.
- [4] Reese, R., Bheda, H., and Mondesir, W., 2016, "Method to Monitor Additive Manufacturing Process for Detection and In-Situ Correction of Defects," Pub. No.: US 2016/0271610 A1 Patent Application Publication.
- [5] Wu, H., Wang, Y., and Yu, Z., 2016, "In Situ Monitoring of FDM Machine Condition via Acoustic Emission," *Int. J. Adv. Manuf. Technol.*, **84**(5–8), pp. 1483–1495.
- [6] Faes, M., Abbeloos, W., Vogeler, F., Valkenaers, H., Coppens, K., Goedemé, T., and Ferraris, E., 2014, "Process Monitoring of Extrusion Based 3D Printing via Laser Scanning," International Conference on Polymers and Moulds Innovations (PMI), Guimarães, Portugal, Sept. 10–12.
- [7] Rao, P. K., Liu, J. P., Roberson, D., and Kong, Z. J., 2015, "Sensor-Based Online Process Fault Detection in Additive Manufacturing," ASME 2015 International Manufacturing Science and Engineering Conference, Guangzhou, China, Nov. 28–29, p. V002T04A010.
- [8] Buffière, J.-Y., Savelli, S., Joineau, P. H., Maire, E., and Fougères, R., 2001, "Experimental Study of Porosity and Its Relation to Fatigue Mechanisms of Model Al–Si7–Mg0.3 Cast Al Alloys," *Mater. Sci. Eng. A*, **316**(1–2), pp. 115–126.
- [9] Guo, Y., Liu, Y., Georgiou, T., and Lew, M. S., 2018, "A Review of Semantic Segmentation Using Deep Neural Networks," *Int. J. Multimed. Inf. Retr.*, **7**(2), pp. 87–93.
- [10] He, K., Gkioxari, G., Dollár, P., and Girshick, R., 2017, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, Oct. 22–29, pp. 2980–2988.
- [11] Pesaresi, M., and Benediktsson, J. A., 2001, "A New Approach for the Morphological Segmentation of High-Resolution Satellite Imagery," *IEEE Trans. Geosci. Remote Sens.*, **39**(2), pp. 309–320.
- [12] Bhatt, P. M., Malhan, R. K., Rajendran, P., Shah, B. C., Thakar, S., Yoon, Y. J., and Gupta, S. K., 2021, "Image-Based Surface Defect Detection Using Deep Learning: A Review," *ASME J. Comput. Inf. Sci. Eng.*, **21**(4), p. 040801.
- [13] Nie, Z., Jiang, H., and Kara, L. B., February 2020, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," *ASME J. Comput. Inf. Sci. Eng.*, **20**(1), p. 011002.
- [14] Ferguson, M., Ak, R., Lee, Y.-T. T., and Law, K. H., 2018, "Detection and Segmentation of Manufacturing Defects With Convolutional Neural Networks and Transfer Learning," *Smart Sustain. Manuf. Syst.*, **2**(1), pp. 137–164.
- [15] Liu, T., Bao, J., Wang, J., and Zhang, Y., 2020, "A Coarse-Grained Regularization Method of Convolutional Kernel for Molten Pool Defect Identification," *ASME J. Comput. Inf. Sci. Eng.*, **20**(2), p. 021005.
- [16] Milletari, F., Navab, N., and Ahmadi, S.-A., 2016, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," IEEE International Conference on 3D Vision, Stanford, CA, Oct. 25–28, pp. 565–571.
- [17] Cicek, O., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O., 2016, "3D U-Net: Learning Dense Volumetric Segmentation From Sparse Annotation," International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Athens, Greece, Oct. 17–21, pp. 424–432.
- [18] Lee, K., Zung, J., Li, P., Jain, V., and Seung, H. S., 2017, "Superhuman Accuracy on the SNEM3D Connectomics Challenge," arXiv Preprint
- [19] Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., and Gulyás, B., 2020, "3D Deep Learning on Medical Images: A Review," *Sensors*, **20**(18), p. 5097.
- [20] Henry, T., Carre, A., Lerousseau, M., Estienne, T., Robert, C., Paragios, N., and Deutsch, E., 2020, "Top 10 BraTS 2020 Challenge Solution: Brain Tumor Segmentation With Self-Ensembled, Deeply-Supervised 3D-Unet Like Neural Networks," arXiv Preprint arXiv:2011.01045.
- [21] Wang, J., Bao, Y., Wen, Y., Lu, H., Luo, H., Xiang, Y., Li, X., Liu, C., and Qian, D., 2020, "Prior-Attention Residual Learning for More Discriminative COVID-19 Screening in CT Images," *IEEE Trans. Med. Imaging*, **39**(8), pp. 2572–2583.
- [22] Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V., 2021, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," *IEEE Access*, **9**, pp. 82031–82057.
- [23] Wong, V. W. H., Ferguson, M., Law, K. H., Lee, Y.-T. T., and Witherell, P., 2020, "Automatic Volumetric Segmentation of Additive Manufacturing Defects With 3D U-Net," AAAI 2020 Spring Symposia, Stanford, CA, Mar. 23–25, arXiv Preprint arXiv:2101.08993.
- [24] Wong, V. W. H., Ferguson, M., Law, K. H., Lee, Y.-T. T., and Witherell, P., 2021, "Segmentation of Additive Manufacturing Defects Using U-Net," Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2021, Virtual, Aug. 17–19.
- [25] Caesar, H., Uijlings, J., and Ferrari, V., 2016, "Region-Based Semantic Segmentation With End-to-End Training," The 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, Oct. 8–16.
- [26] Long, J., Shelhamer, E., and Darrell, T., 2015, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, June 7–12, pp. 3431–3440.
- [27] Ronneberger, O., Fischer, P., and Brox, T., 2015, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Munich, Germany, Oct. 5–9, pp. 234–241.
- [28] Yu, Q., Xia, Y., Xie, L., Fishman, E. K., and Yuille, A. L., 2019, "Thickened 2D Networks for 3D Medical Image Segmentation," arXiv Preprint arXiv 1904.01150.
- [29] Isensee, F., Jaeger, P. F., Full, P. M., Wolf, I., Engelhardt, S., and Maier-Hein, K. H., 2017, "Automatic Cardiac Disease Assessment on Cine-MRI via Time-Series Segmentation and Domain Specific Features," *The 8th International Statistical Atlases and Computational Modeling of the Heart (STACOM) Workshop*, Quebec City, Canada, Sept. 10–14, pp. 120–129.
- [30] Zhang, B., Liu, S., and Shin, Y. C., 2019, "In-Process Monitoring of Porosity During Laser Additive Manufacturing Process," *Addit. Manuf.*, **28**, pp. 497–505.
- [31] Shevchik, S. A., Kenel, C., Leinenbach, C., and Wasmer, K., 2018, "Acoustic Emission for in Situ Quality Monitoring in Additive Manufacturing Using Spectral Convolutional Neural Networks," *Addit. Manuf.*, **21**, pp. 598–604.
- [32] Mutiargo, B., Pavlovic, M., Malcolm, A. A., Goh, B., Krishnan, M., Shota, T., Shaista, H., Jhinaoui, A., and Putro, M. I. S., 2019, "Evaluation of X-Ray Computed Tomography (CT) Images of Additively Manufactured Components Using Deep Learning," 3rd Singapore International Non-Destructive Testing Conference and Exhibition (SINCE2019), Singapore, Dec. 4–5.
- [33] Kim, F. H., Moylan, S. P., Garboczi, E. J., and Slotwinski, J. A., 2017, "Investigation of Pore Structure in Cobalt Chrome Additively Manufactured Parts Using X-Ray Computed Tomography and Three-Dimensional Image Analysis," *Addit. Manuf.*, **17**, pp. 23–38.
- [34] Kim, F. H., Moylan, S. P., Garboczi, E. J., and Slotwinski, J. A., 2019, *High-Resolution X-Ray Computed Tomography (XCT) Image Data Set of Additively Manufactured Cobalt Chrome Samples Produced With Varying Laser Powder Bed Fusion Processing Parameters, CoCr AM XCT Data*, National Institute of Standards and Technology, Gaithersburg, MD.
- [35] Buades, A., Coll, B., and Morel, J.-M., 2011, "Non-Local Means Denoising," *Image Process. Line*, **1**, pp. 208–212.
- [36] Sun, W., Brown, S. B., and Leach, R. K., 2012, *An Overview of Industrial X-Ray Computed Tomography, Technical Report ENG 32*, National Physical Laboratory, Teddington, Middlesex, UK.
- [37] Bernsen, J., 1986, "Dynamic Thresholding of Gray-Level Images," 8th International Conference on Pattern Recognition, Paris, France, Oct. 27–31, pp. 1251–1255.
- [38] Ioffe, S., and Szegedy, C., 2015, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 32nd International Conference on Machine Learning, Lille, France, July 6–11.
- [39] Wu, Y., and He, K., 2020, "Group Normalization," *Int. J. Comput. Vis.*, **128**, pp. 742–755.
- [40] Ulyanov, D., Vedaldi, A., and Lempitsky, V., 2016, "Instance Normalization: The Missing Ingredient for Fast Stylization," arXiv Preprint arXiv:1607.08022.
- [41] Werbos, P. J., 1990, "Backpropagation Through Time: What It Does and How to Do It," *Proc. IEEE*, **78**(10), pp. 1550–1560.
- [42] Rakhlin, A., Davydov, A., and Nikolenko, S. I., 2018, "Land Cover Classification From Satellite Imagery With U-Net and Lovasz-Softmax Loss," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, June 18–22.
- [43] Diakogiannis, F. I., Waldner, F., Caccetta, P., and Wu, C., 2020, "ResUNet-a: A Deep Learning Framework for Semantic Segmentation of Remotely Sensed Data," *ISPRS J. Photogramm. Remote Sens.*, **162**, pp. 94–114.
- [44] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 26–July 1, pp. 770–778.
- [45] Hinton, G., 2012, "Neural Networks for Machine Learning—Lecture 6a—Overview of Mini-Batch Gradient Descent," [https://www.cs.toronto.edu/~tjmen/csc321/slides/lecture\\_slides Lec6.pdf](https://www.cs.toronto.edu/~tjmen/csc321/slides/lecture_slides Lec6.pdf), Accessed July 2020.
- [46] GitHub, Inc., "UNet: Semantic Segmentation With PyTorch," <https://github.com/milesial/Pytorch-UNet>, Accessed November 2020.
- [47] Kingma, D., and Ba, J., 2015, "Adam: A Method for Stochastic Optimization," 3rd International Conference for Learning Representations (ICLR), San Diego, CA, May 7–9.
- [48] Wolny, A., 2019, "Pytorch-3DUnet: PyTorch Implementation of 3D U-Net," Zenodo.
- [49] Isensee, F., Jäger, P. F., Kohl, S. A. A., Petersen, J., and Maier-Hein, K. H., 2020, "nnU-Net: A Self-Configuring Method For Deep Learning-Based Biomedical Image Segmentation," *Nature Methods*, **18**, pp. 203–211.
- [50] Drozdal, M., Vorontsov, E., Chartrand, G., Kadoury, S., and Pal, C., 2016, "The Importance of Skip Connections in Biomedical Image Segmentation," *Deep Learn. Data Labeling Med. Appl.*, **10008**, pp. 179–187.
- [51] Khened, M., Kollerathu, V. A., and Krishnamurthi, G., 2019, "Fully Convolutional Multi-Scale Residual DenseNets for Cardiac Segmentation and Automated Cardiac Diagnosis Using Ensemble of Classifiers," *Med. Image Anal.*, **51**, pp. 21–45.
- [52] He, K., Zhang, X., Ren, S., and Sun, J., 2015, "Delving Deep Into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, Dec. 7–13, pp. 1026–1034.
- [53] Yurii, N., 2004, *Introductory Lectures on Convex Optimization: A Basic Course*, 1, Springer Science & Business Media, Boston, MA.
- [54] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D., 2018, "Attention U-Net: Learning Where to Look for the Pancreas," *Medical Imaging with Deep Learning (MIDL)*, Amsterdam, The Netherlands, July 4–6.