**Proceedings of the ASME 2021**
**International Design Engineering Technical Conferences and**
**Computers and Information in Engineering Conference**
**IDETC/CIE2021**
**August 17-20, 2021, Virtual, Online**

**DETC2021-XXXX**

# DRAFT: SEGMENTATION OF ADDITIVE MANUFACTURING DEFECTS USING U-NET

**Vivian Wen Hui Wong[1], Max Ferguson[1], Kincho H. Law[1], Yung-Tsun Tina Lee[2], and Paul Witherell[2]**

[1]Engineering Informatics Group, Civil and Environmental Engineering, Stanford University, Stanford, CA
[2]Systems Integration Division, National Institute of Standards and Technology (NIST), Gaithersburg, MD

## ABSTRACT

*Additive manufacturing (AM) provides design flexibility and allows rapid fabrications of parts with complex geometries. The presence of internal defects, however, can lead to deficit performance of the fabricated part. X-ray Computed Tomography (XCT) is a non-destructive 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 paper 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. 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 work demonstrates that, using XCT images, U-Net can be effectively applied for automatic segmentation of AM porosity with high accuracy. The method can potentially help improve quality control of AM parts in an industry setting.*

Keywords: Smart Manufacturing, Defect Detection, Additive Manufacturing, Convolutional Neural Networks

## 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 manufacturing of parts with complex geometries [1,2]. The fabrication process, however, comes with increased 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 [3]. The ability to automatically identify defects of parts fabricated using AM is essential.

Current trends with non-destructive 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 [4,5,6]. 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 properties of a completed AM build. It can be used to visualize internal structures and identify small pores and flaws in an AM part [7]. 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 [8], and have been shown effective in many domains, including everyday objects [9], satellite imagery [10], and metal casting defects in manufacturing [11]. 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 images [12]. 3D CNNs have demonstrated potential in volumetric medical image segmentation [12,13,14]. Among existing 3D CNN methods, those with an encoder-decoder based architecture, also known as U-Net variants, have achieved excellent performance in several medical image segmentation tasks with relatively small number of training samples [15] and are increasingly popular in medical image segmentation applications, such as brain tumor segmentation [16] and 3D chest CT image segmentation for COVID-19 screening [17]. 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 between 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 public dataset poses a huge challenge to the adoption of machine-learning approaches, which require a large number of training data to converge on a reasonable model. Despite these challenges, the mainstream adoption of machine learning methods for defect detection of AM parts is essential to the developing 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 3D U-Net model with existing defect labels to automatically segment defects in XCT images of unknown AM samples [18]. In this paper, we focus on AM defects such as pores and cracks, or any internal, possibly defect-indicative, volumetric voids in the part. We show that 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 work 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 rest of the paper is organized as follows: Section 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, the paper is concluded with a brief summary and discussion in Section 6.

## 2. RELATED WORKS

With processing, 3D images can be sliced into 2D and vice versa, thereby allowing 2D CNNs to segment volumetric images [12]. The most commonly used CNN architectures for 2D segmentation problem are region-based and fully-convolutional-network-based (FCN-based) [8]. Region-based CNN (R-CNN), such as the Mask R-CNN, is an example of the former [8,9]. Since regions need to first be extracted, described, then classified, these methods are generally more computationally expensive [19]. On the other hand, FCN-based methods directly learn a mapping from input to output pixels, without proposing regions [20]. U-Net is a CNN model that extends the FCN architecture, achieving excellent performance, for example, in the segmentation of ventral nerve cord [21].

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 prior to training loses information on the correlation between slices [22]. To that end, 3D FCN-based segmentation architectures, such as 3D U-Net [13] and V-Net [12], 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 pre-trained models, leading to less stable training [22]. 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 is anisotropic [23]. 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 [24]. CNN has also been used to analyze acoustic emissions during AM processes [25]. More recently, 2D U-Net has been deployed to a dataset similar to that used in this work and achieved a mean IOU of 0.92 [26]. This paper not only presents the novel CNN techniques for porosity segmentation, but also 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. [27] and is publicly available [28]. 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 (LPBF). Artificial pores and cracks were produced by changing AM scan speed and hatch spacing. By changing the process parameters, specimens were processed

**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 |

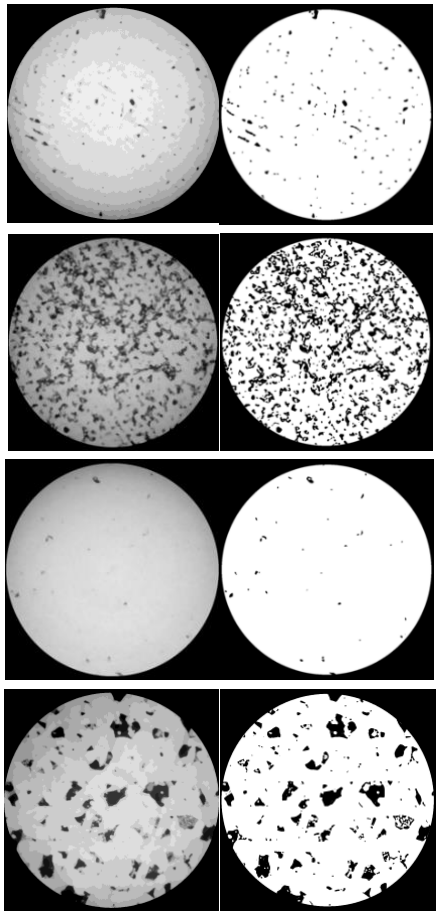


**FIGURE 1:** EXAMPLES OF IMAGES FROM AM DEFECT DATASETS: PROCESSED XCT IMAGES ARE ON THE LEFT AND SEGMENTATION MASKS ARE ON THE RIGHT

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 non-local means filter [29,30]. To obtain ground truth labeling of the defects, Bernsen local thresholding [31] was used to process the 8-bit images. The local contrast threshold parameters of the thresholding process are computed by relating average noise value to local contrast threshold as explained by Kim et al. [27]. Figure 1 shows examples of images and 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.

In order 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 Figure 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 distribution of pores may be **anisotropic** [27].
- 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 Figure 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 images are available for the study. Three specimens (samples 2, 3 and 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 (CNNS) 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.

### 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 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 3 directions, along all 3 axes of the input image and its feature maps. A 2D kernel moves in 2 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 $\boldsymbol{\theta}^{(l)}$:

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

where $h^{(l)}$ is the output feature map of layer $l$ and $h^{(0)}$ is an

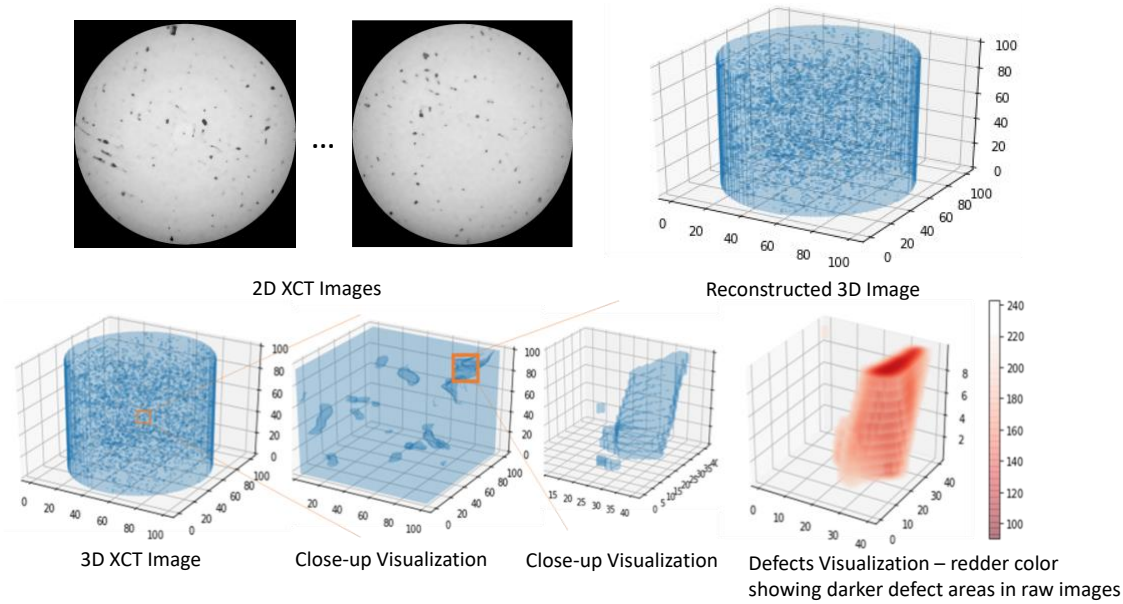



**FIGURE 2:** RECONSTRUCTION OF A 3D AM IMAGE FROM 2D XCT IMAGES.

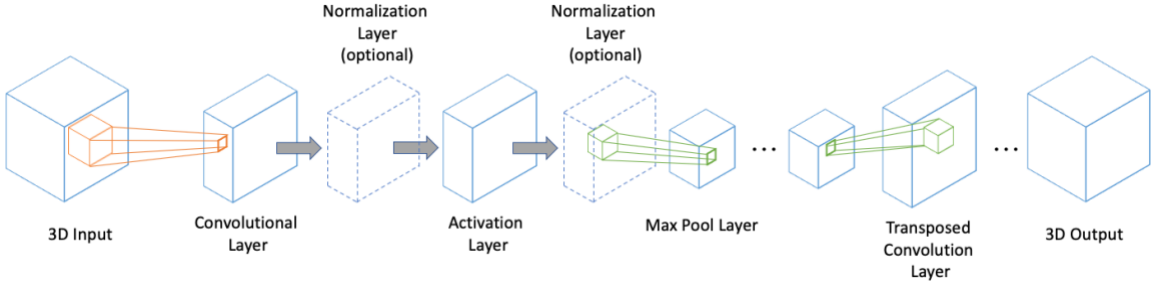



**FIGURE 3:** EXAMPLE OF A 3D CNN BUILT BY COMBINING VARIOUS TYPES OF LAYERS

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:

$$h^{(l)} = \sigma\left(f^{(l)}\left(h^{(l-1)}; \boldsymbol{\theta}^{(l)}\right)\right) \tag{2}$$

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 in order to improve the speed and stability of training neural networks [32]. 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 [33]. Instance normalization (IN) is another technique that normalizes across spatial locations [34].

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, $\boldsymbol{\theta} = \{\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^L\}$, 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 [35].

### 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 [16,17,21] and but also in non-medical domains such as satellite imagery [36]. 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

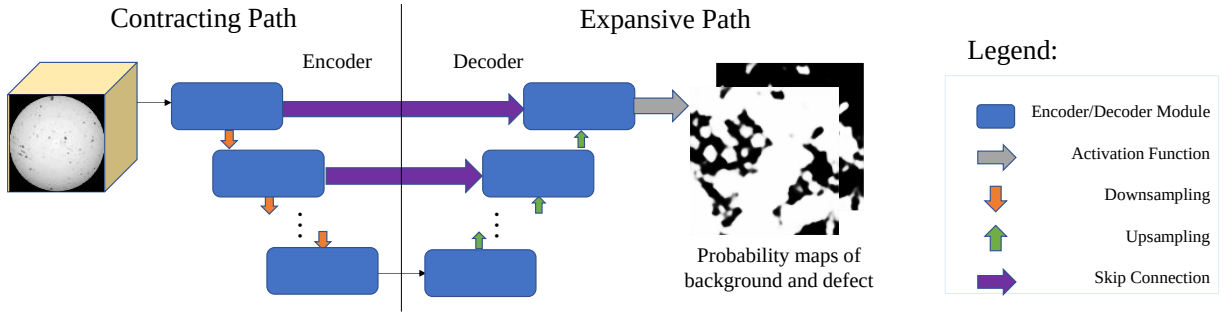



**FIGURE 4:** SCHEMATIC OF A GENERAL U-NET ARCHITECTURE

**TABLE 2:** DETAILS OF U-NET BASE CONFIGURATION

| Model | Encoder/Decoder | Downsampling | Upsampling |
|---|---|---|---|
| 2D U-Net | (3 × 3 convolution +BN + ReLU) × 2 | 2 × 2 max pool | Bilinear with a scale of 2 |
| Vanilla 3D U-Net | (3 × 3 × 3 convolution +BN + ReLU) × 2 | 2 × 2 × 2 max pool | 2 × 2 × 2 transposed convolution |
| 3D U-Net with GN | (3 × 3 × 3 convolution +ReLU+ GN) × 2 | | |
| Residual 3D U-Net | (3 × 3 × 3 convolution +GN+ ReLU) × 3 | | |

encoders and decoders. Figure 4 shows the overall structure of a U-Net architecture. Implementation in this work 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 [21]. On the other hand, 3D U-Net, described in [13], deploys 3D convolutions and adds batch normalization to its encoder and decoder modules. Furthermore, the ResUNet-a presented in [37] discovered that residual connections could improve the performance of U-Net and can help reduce the vanishing gradient problem [38]. 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 [14,38].

### 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 2D U-Net follows the same implementation as described in [21]. The model's encoder and decoder modules are each a double convolution: twice stacking a 2D convolutional layer, followed by a ReLU activation. 3×3 kernels are employed for the convolutional layers, and 2×2 max pooling is used for downsampling. Some minor modifications from the standard implementation are made: First, bilinear upsamplings are used instead of transposed convolution to save memory. Furthermore, the 2D convolutions are zero padded with a one-pixel border to preserve the features of the edges. Lastly, 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 a RMSprop optimizer [39], and a learning rate of 0.0001 is used. The 2D U-Net implementation is adapted from a publicly available PyTorch implementation [40].

The vanilla 3D U-Net, on the other hand, follows the implementation by Cicek et al. [13]. 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. [14], 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 Section 3. Due to memory constraint, each training sample is a 128×128×128 patch randomly sampled from the 3D image. Stride sizes are 32×32×32 to overlap the patches and ensure that information is not lost. The input patches are normalized, randomly flipped and rotated prior to training. Network outputs and targets are compared using the cross-entropy 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 [41]. 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 [42]. All models are trained on NVIDIA Tesla T4 GPUs with 100 GB RAM and 4 Intel virtual CPU on Google Cloud Platform.

### 4.4 EXPERIMENTAL RESULTS

The prediction accuracy of each of the above-mentioned models is evaluated using the mean IOU metric, comparing the accuracy of a predicted segmentation with the ground truth or labeled mask. Table 3 shows 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 [20], 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.

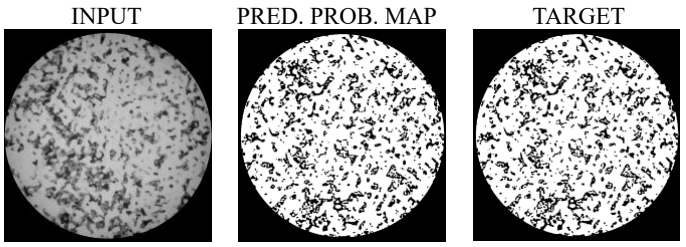
We observe several limitations posed by the 3D models. Figure 5 shows an example patch sampled by the Residual 3D U-Net from the validation data. It can be seen that due to the large size of the images, the $128 \times 128 \times 128$ patches only capture the shape of defects partially. This could limit 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 Figure 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 Figure 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 Figure 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.
4) **High resolution:** The resolution size of the input images in the referenced AM dataset is very large. Table 1 shows the number of voxels and the array shape of each specimen image. This not only leads to a high memory consumption during preprocessing and training, but also discards the possibility to conduct downsampling prior to training, as downsampling would lose valuable information on the already rare small defects.

**TABLE 3:** VALIDATION MEAN IOU AND AVERAGE TRAINING TIME PER EPOCH ON THE BASE U-NET MODELS

| Model | Training Time (hours) | Validation mean IOU |
|---|---|---|
| 2D U-Net | **0.70** | **0.993** |
| 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 |



(a) EXAMPLE 2D IMAGE OUTPUTTED BY 2D U-NET


(b) EXAMPLE PATCH OF A DEFECT OUPUTTED BY RESIDUE 3D U-NET

**FIGURE 5:** EXAMPLES OF SEGMENTATION RESULTS OUTPUTTED BY 2D U-NET AND RESIDUE 3D U-NET

In the next section, 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 post-processing for image segmentation [43]. 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 Figure 6 are as follows:

- **Nonzero Cropping:** The images are cropped into regions where voxel values are nonzero. Each AM specimen image consists of a large number of voxels. Such large arrays of voxels induce a heavy computational cost. Cropping the images 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.

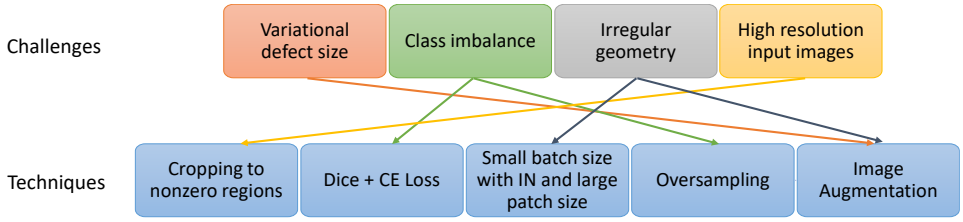



**FIGURE 6:** TECHNIQUES USED TO ADDRESS SEGMENTATION CHALLENGES POSED BY AM DEFECTS

- **Dice + Cross-Entropy (CE) Loss:** A loss function that sums the Dice loss and cross-entropy (CE) loss is used [44]. 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 true Dice score and lead to unstable training. On the other hand, cross-entropy (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 [45]. 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 cross-entropy (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 are 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. For this reason, the batch size is selected to be small. While batch normalization [32] is often used in CNN training to improve robustness and convergence, but, because of the smaller batch size, instance normalization [34] is used. Lastly, the size of kernels is calculated by limiting the total size of feature maps to the GPU memory budget.
- **Oversampling:** Since we have less voxels of defects than voxels of background, the 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 the network to generalize defects with various aspect ratios, colors, and shapes. Details and specifics of these augmentation techniques have been described by Isensee et al. on nnU-Net [43].

In addition to the data enhancements made above, 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 [43].

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 shown 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. [46]. Stochastic gradient descent with Nesterov momentum [47] 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

**TABLE 4:** DETAILS OF ENHANCED U-NET CONFIGURATION

| Model | Encoder/Decoder | Downsampling | Upsampling |
|---|---|---|---|
| 2D U-Net (Patched) | $(3 \times 3$, stride 2 convolution + IN + Leaky ReLU$) \times 2$ | Done through strided convolution | $2 \times 2$ transposed convolution |
| 3D U-Net | $(3 \times 3 \times 3$, stride 2 convolution +IN + Leaky ReLU$) \times 2$ | | $2 \times 2 \times 2$ transposed convolution |
| Residual 3D U-Net | $(3 \times 3 \times 3$, stride 2 convolution +IN + Leaky ReLU$) \times 2$ | | |

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 GPU's capacity, 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 shown 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 non-patched 2D U-Net as shown 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 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 paper presents the use of U-Net models for automatic detection of AM defects using XCT images. 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 Section 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 un-intended porosity, for example, for parts that have internal features such as holes and channels inside.

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

**TABLE 5:** VALIDATION MEAN IOU AND AVERAGE TRAINING TIME PER EPOCH ON ENHANCED U-NET MODELS

| Model | Training Time (hours) | Validation mean IOU |
|---|---|---|
| 2D U-Net (Patched) | 4.55 | 0.988 |
| 3D U-Net | 21.06 | 0.992 |
| Residual 3D U-Net | 20.61 | **0.993** |

tailored towards 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 [48].

In summary, while conventional manual or thresholding methods for AM defect segmentation remain tedious and unscalable, this paper 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.


## ACKNOWLEDGEMENTS

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.


## 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

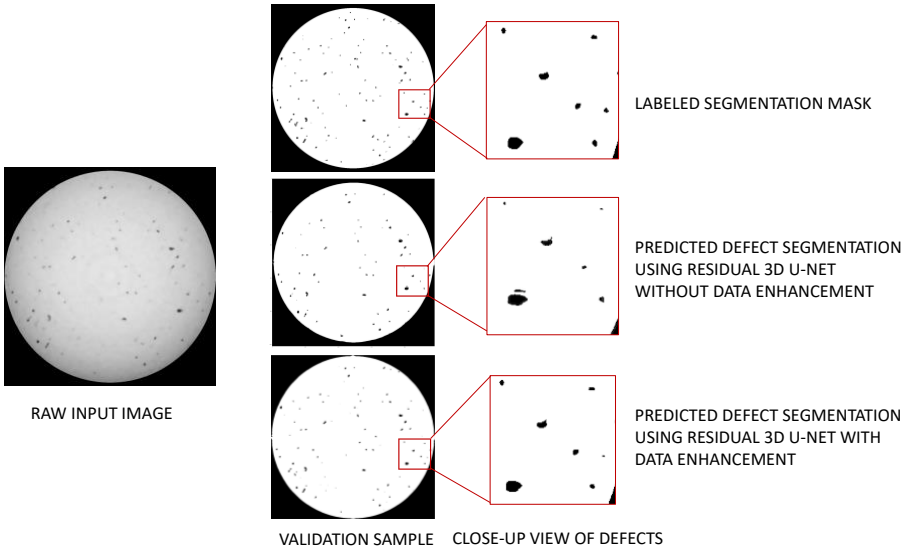



**FIGURE 7:** 2D SLICE OF A SAMPLE SEGMENTED BY RESIDUAL 3D U-NET WITH AND WITHOUT DATA ENHANCEMENT


review of materials, methods, applications and challenges," *Composites Part B: Engineering*, **143**, pp.172–196.

[3] 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*.

[4] Wu, H., Wang, Y., and Yu, Z., 2016, "In situ monitoring of FDM machine condition via acoustic emission," *International Journal of Advanced Manufacturing Technology*, **84**(5-8), pp. 1483-1495.

[5] Faes, M., Abbeloos, W., Vogeler, F., Valkenaers, H., Coppens, K., Goedemé, T., and Ferraris, E., 2016, "Process monitoring of extrusion based 3D printing via laser scanning," *arXiv preprint arXiv:1612.02219*.

[6] 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*, ASME Digital Collection, V002T04A010-V002T04A010.

[7] Buffiere, J.-Y., Savelli, S., Jouneau, 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," *Materials Science and Engineering: A*, **316**(1–2): pp. 115–126.

[8] Guo, Y., Liu, Y., Georgiou, T., and Lew, M. S., 2018, "A review of semantic segmentation using deep neural networks," *International Journal of Multimedia Information Retrieval*, **7***(*2): pp. 87–93.

[9] He, K., Gkioxari, G., Dollár, P., and Girshick, R., 2017, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Venice, Italy, pp. 2980–2988.

[10] Pesaresi, M., and Benediktsson, J. A., 2001, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, **39**(2), pp. 309-320.

[11] 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 and Sustainable Manufacturing Systems,* **2**(1), pp. 137-164.

[12] Milletari, F., Navab, N., and Ahmadi, S.-A., 2016, "V-Net: fully convolutional neural networks for volumetric medical image segmentation," *IEEE International Conference on 3DVision*, pp. 565-571.

[13] 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)*, pp. 424–432.

[14] Lee, K., Zung, J., Li, P., Jain, V., and Seung, H. S., 2017, "Superhuman accuracy on the SNEMI3D Connectomics challenge," *arXiv preprint arXiv:1706.00120*,.

[15] 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), pp.5097.

[16] 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*.

[17] 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 Transactions on Medical Imaging*, **39**(8), pp.2572-2583.

[18] 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, USA, Mar 23-25, 2020. *arXiv preprint arXiv:2101.08993.*

[19] Caesar, H., Uijlings, J., and Ferrari, V., 2016, "Region-based semantic segmentation with end-to-end training," *arXiv preprint arXiv 1607.07671.*

[20] Long, J., Shelhamer, E., and Darrell, T., 2015, "Fully convolutional networks for semantic segmentation," *IEEE*

*Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440.

[21] 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)*, pp. 234–241.

[22] 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.*

[23] 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," *International Workshop on Statistical Atlases and Computational Models of the Heart*, pp. 120-129.

[24] Zhang, B., Liu, S., and Shin, Y. C., 2019, "In-process monitoring of porosity during laser additive manufacturing process," *Additive Manufacturing*, **28**, pp. 497–505.

[25] 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," *Additive Manufacturing*, **21**, pp. 598–604.

[26] 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).*

[27] Kim, F.H., Moylan, S.P., Garboczi, E.J., Slotwinski, J.A., 2017, "Investigation of pore structure in cobalt chrome additively manufactured parts using X-Ray computed tomography and three-dimensional image analysis," *Additive Manufacturing*, **17**, pp. 23-38.

[28] Kim, F.H., Moylan, S.P., Garboczi, E.J., 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. Available at https://doi.org/10.18434/M32162. (Accessed 11/2019).

[29] Buades, A., Coll, B., and Morel, J-M., 2011, "Non-local means denoising," *Image Processing On Line*, **1**, pp. 208-212.

[30] 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, United Kingdom.

[31] Bernsen, J., 1986, "Dynamic thresholding of gray-level images," *8th International. Conference on Pattern Recognition*, pp. 1251–1255.

[32] Ioffe, S., and Szegedy, C., 2015, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167.*

[33] Wu, Y., and He, K., 2018, "Group normalization," *arXiv preprint arXiv:1803.08494*

[34] Ulyanov, D., Vedaldi, A., and Lempitsky, V., 2016, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*.

[35] Werbos, P. J., 1990, "Backpropagation through time: what it does and how to do it," **78**, pp. 1550–1560.

[36] Rakhlin, A., Davydow, A., and Nikolenko, S.I., 2018, "Land cover classification from satellite imagery with U-Net and Lovasz-softmax loss," *CVPR Workshops*, pp. 262-266.

[37] 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 Journal of Photogrammetry and Remote Sensing*, **162**, pp. 94-114.

[38] 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)*, pp. 770–778.

[39] Hinton, G., 2012, *Neural Networks for Machine Learning - Lecture 6a - Overview of Mini-Batch Gradient Descent*. Available at https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. (Accessed 7/2020)

[40] *UNet: Semantic Segmentation with PyTorch*, Available at https://github.com/milesial/Pytorch-UNet (Accessed 11/2020).

[41] Kingma, D., and Ba, J., 2014, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980.*

[42] Wolny, A., 2019, "Wolny/Pytorch-3DUnet: PyTorch implementation of 3D U-Net," *Zenodo*. http://doi.org/10.5281/ zenodo.2671581

[43] Isensee, F., Jäger, P.F., Kohl, S.A.A., Petersen, J., and Maier-Hein, K.H., 2020, "Automated design of deep learning methods for biomedical image segmentation," *arXiv preprint arXiv:1904.08128*.

[44] Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., and Pal, C., 2016, "The importance of skip connections in biomedical image segmentation," *Deep Learning and Data Labeling for Medical Applications*, pp. 179-187.

[45] 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," *Medical image analysis*, *51*, pp.21-45.

[46] 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*, pp. 1026-1034.

[47] Yurii, N., 2013, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Science & Business Media.

[48] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., and Glocker, B., 2018, "Attention U-Net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*.