Unsupervised Defect Classification of 2D SEM and 3D X-Ray CT Images from Laser Powder Bed Fusion

Andrew Lang[1,3], Cesar Ortiz Rios[2], Joseph Newkirk[2], Robert G. Landers[3], James Castle[1], and Douglas A. Bristow[3]

[1]The Boeing Company
[2]Materials Science and Engineering, Missouri University of Science and Technology
[3]Mechanical and Aerospace Engineering, Missouri University of Science and Technology

Abstract

This work discusses a method to classify defects in laser powder bed fusion using 2D images of layer samples taken by Scanning Electron Microscope (SEM) and 3D image stacks of a full part by X-Ray Computed Tomography (XCT). Images using SEM are taken of a sampled layer in a printed part and unsupervised classification of defects in the SEM images is performed with Otsu's thresholding method, K-means classification, and the Robust Automatic Threshold Selection algorithm. The performance of the classifiers, measured against human-generated ground truth defect labels, is improved by registering and fusing multiple SEM images taken under different settings and detector locations. Otsu's method is shown to be the best classifier for the 3D XCT dataset. Finally, the 2D sample is located in the 3D XCT array and the reliability of the 3D defect classification technique is validated.

1.0 Introduction

Laser Powder Bed Fusion (LPBF) is a form of Additive Manufacturing where a laser selectively fuses consecutive layers of metal powder to additively build components [1]. There is increasing interest in LPBF in several industries, including aerospace and medical, as well as academic research due to increased freedom of design and cost advantages for small production runs [2, 3]. This study outlines a process to create a 3D profile of a part printed by LPBF by classifying anomalous vs nominal voxels in the part for use as ground truth labels in machine learning. Also, a method is outlined to locate an extracted 2D section in a 3D XCT array. This classification method is intended to be used to streamline machine learning training and not to make Quality Assurance (QA) accept/reject decisions although it may support QA analysis. The process developed is transferrable to other data modes and scalable to larger datasets as implementation is performed with open-source and free toolkits rather than proprietary tools or systems.

X-ray Computed Tomography (XCT) has become standard in industrial verification of additively manufactured (AM) parts [4]. XCT is advantageous for quality control because it is non-destructive and, depending on the resolution required, can relatively quickly capture profiles of a whole part. XCT is commonly used to detect pores which is critically important for many applications where failure is dominated by individual flaws [5]. The flaw detection limit from XCT has been estimated to be 2-3 voxels in diameter [4, 6]. Various studies have been done using XCT to determine porosity and correlate with data from in situ sensors [7, 8, 9]. This work can be used

to support XCT classification to fuse with in situ process data and make predictions of follow-on builds.

Material characterization, using micrographs from Scanning Electron Microscopes (SEM), is often used to support qualification of LPBF processes but the part must be destroyed to use SEM and only samples of the part can be taken [10, 11]. SEM has been shown to be an effective tool, alongside XCT, to analyze large in homogeneous void distributions [5]. An unsupervised method to improve SEM classification by fusing multiple SEM imaging positions together is outlined in Section 4.3.

Other studies have classified XCT data using a human expert to interrogate the data [6], by manually defining a classification threshold [7, 8, 12], or by using Otsu's method to automatically compute a threshold for binary classification [9]. Internal flaws are classified in this work using a combination of XCT 3D array of a full part printed by LPBF and 2D SEM images of a destructed sample in the part. K-means and Otsu's method are each used to automatically select a grayscale threshold and segregate pixels (for the 2D SEM images) and voxels (for the 3D XCT array) into classes. Classification of a sampled layer, found by SEM, is used to verify the 3D profile classification of the XCT data array.

This work uses DREAM.3D to implement segmentation tools in the Insight Toolkit (ITK) to identify and classify image data from samples created by LPBF. ITK is an open-source software toolkit for performing registration and segmentation of image data [13] and was developed by the National Library of Medicine and several partners to support the creation of a public resource in high-dimension data processing tools and image analysis [14]. DREAM.3D is an open-source tool kit that allows for construction of customized workflows to analyze data [15]. DREAM.3D was developed to process digital instances of microstructure, but many of the tools are useful for LPBF data processing. ParaView, an open-source analysis and visualization platform, is used to render 2D and 3D visualizations (paraview.org).The main advantages of DREAM.3D are that the reconstruction, manipulation, and visualization of 3D data is simple and straightforward, the program is free and open-source, over 100 filters from the image processing library ITK are included, data is stored as standard HDF5 files by default, additional features are continuously added, and custom filters can be created.

2.0 Experimental Setup

For this study, an ASTM E8 tensile specimen [16] was printed using 304L stainless steel powder in a Renishaw AM250. The test specimen is 50 mm tall and has a 4 mm diameter neck. The nominal laser power for this build was 200W and a section of lettering ("Missouri S&T") was printed at 100W inside the neck of the part to induce defects. The point distance, d_p, and the hatch spacing, d_h, were held at constant 60 μm and 85 μm, respectively. SEM images were taken of a 2D extracted section of the test specimen using multiple imaging positions and XCT imaging was used to generate a 3D array of the part density.

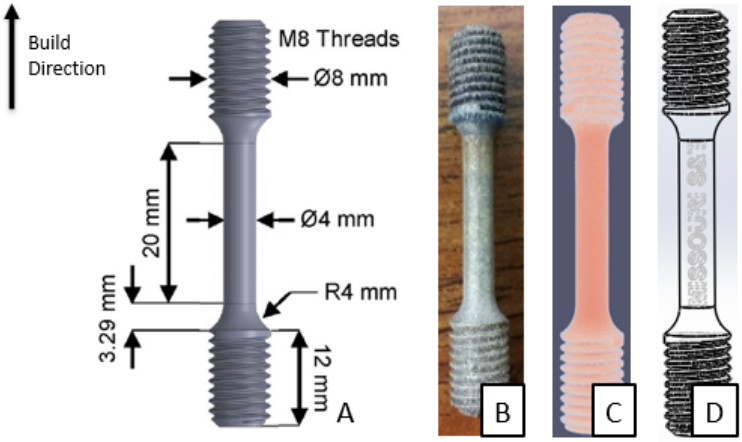


Figure 1: ASTM E8 tensile specimen (a), the tensile bar printed by LPBF (b), 3D reconstruction of XCT imaging (c), and CAD model showing lettering section printed at lower power (d).

2.1 SEM Imaging

The tensile bar was cut perpendicular to the long axis via wire EDM as shown in the cut plan in Figure 2 (a). The "M" section in the text printed at lower power was extracted between 36.65 and 31.65 mm and mounted in conductive mounting resin then metallographically prepared to a final polish of 0.05 µm for SEM analysis. During the metallographic preparation, 331 µm of material was removed, resulting in a cross section containing the targeted "M" low power region. Imaging parameters were selected to generate good contrast between the polished surface metal and the defect interior to facilitate registration in DREAM.3D. Images were captured with the Through-Lens Detector (TLD) in Backscatter Electron (BSE) mode biased to -150V voltage to eliminate the associated secondary electron (SE) edge effect. Integration of eight frames with a dwell time of 1 µs were used. The orientation of the 0º imaging position was chosen to align to the orientation of the CT data. Images were taken in a 5x5 grid pattern starting from row 3, column 1 in increments of 1000 µm in the X-direction and 800 µm in the Y-direction. Increments were chosen to ensure sufficient overlap for stitching the images together. Once the images were captured, the stage was rotated to establish a new imaging position. The imaging process was repeated until the eight imaging positions, shown in Figure 2 (b), were captured. An image manipulation program was used to stitch the images together. The images were imported as layers and overlayed using the overlapping features. Layers were ordered such that lower rows overlapped higher rows and interior columns overlapped exterior columns. For example, row 1 is overlapped by row 2, and column 3 overlaps both columns 2 and 4, which overlap columns 1 and 5, respectively. Pixel sizes of these images are 1.25 µm x 1.25 µm.

When the images were arranged, the image was flattened and exported as a TIF file. The images were integrated into DREAM.3D using the *ITK:Image Reader* filter. The *Set Origin & Spacing (Image)* filter was used to input the spacing based the pixel dimensions.

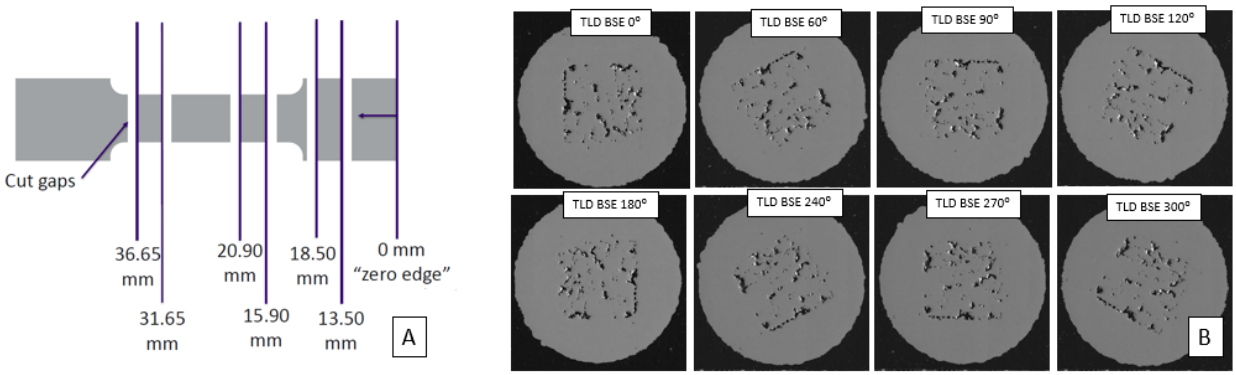


Figure 2 Extracted section for SEM imaging: Cut Plan (a), 8 SEM imaging positions (b)

2.2 X-Ray Computed Tomography Imaging

Post-process XCT inspection was performed by NSI using a custom radiograph tool. NSI scanned the test piece with a MeVX6™ High Energy system using a 6 MeV linear accelerator (LINAC) at 450 kV and performed 360º step scan imaging with a focal spot size of 24.15 µm. The full geometry of the tensile piece was captured in 291 x 281 x 1706 voxels. Each voxel is a cube with dimensions 29.96 µm in X, Y, and Z. The XCT array was directly imported into DREAM.3D using the *Import North Star Imaging CT (.sihdr/.nsidat)* filter.

3.0 SEM and XCT Classification

Pixels, in each SEM image, and voxels, in the XCT array, were classified to represent nominal and anomalous regions. Binary classification is done by selecting a threshold to segregate data points using the Robust Automatic Threshold Selection (RATS), Otsu's method, and K-means classification.

3.1 RATS Classification

The 2D SEM images were classified using the RATS algorithm, an automatic thresholding method based on gradients in the image. The 3D XCT array was not classified by RATS for reasons discussed in section 3.1.2. RATS is an automated edge detector algorithm that identifies a threshold for greyscale images based on the image's gradients, often at the maximum gradient [17, 18]. The gradient array was calculated using the *ITK::Gradient Magnitude Image Filter*. The magnitude of the image gradient is:

$$G(x,y) = \|f(x,y)\|_2 = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \quad [1]$$

The direction of the image gradient is:

$$\theta(x,y) = \tan^{-1}\frac{f_x(x,y)}{f_y(x,y)} \quad [2]$$

Eliminating the square root of the magnitude yields suitable results without the added cost of a final scan across the image to compute the root. In the ITK plugin, the gradient is simply the sum of the squares of the partial derivative operations.

$$g(x,y) = G^2_{(x,y)} = f_x(x,y)^2 + f_y(x,y)^2 \quad [3]$$

With the *Robust Automatic Threshold* filter, DREAM.3D creates a new array that is false where the input array is less than the threshold and true otherwise. A regional threshold is computed as the gradient weight sum of the input array, A.

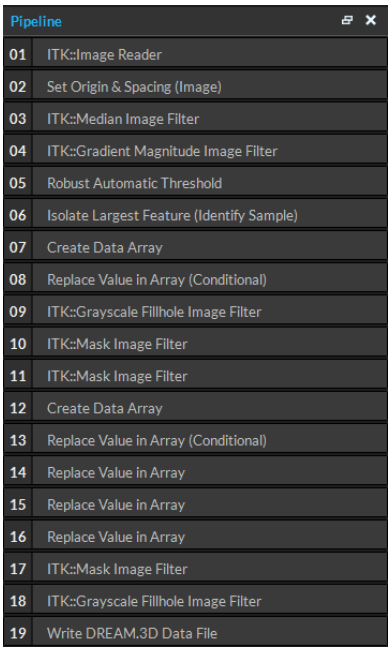
$$T = \frac{\sum_{i=1}^{n}(g(x,y)\cdot A(x,y))}{\sum_{i=1}^{n} g(x,y)} \quad [4]$$

This threshold is commonly at local maxima of the gradient magnitude.

The *ITK::Grayscale Fillhole Image* filter, with the FullyConnected parameter set to true, is used to isolate and fill the part mask.

3.1.1 SEM with RATS

The pipeline on the left and the images in Figure 3 demonstrate the process to classify each of the 8 SEM images with the RATS threshold. The individual SEM images were read into DREAM.3D using the *ITK::Image Reader* (step 1 and Fig 3a). *Set Origin & Spacing (Image)* was used to scale the pixel size to normal units of 0.00125 mm per voxel in both the X and Y dimensions (step 2). ITK filters were used to create a median array of radius 1 pixels and calculate the gradient magnitude of the median array (steps 3, 4 and Figure 3b). The RATS algorithm was implemented to automatically threshold the image array along gradient lines and *Isolate Largest Feature* was used to remove features outside the part boundary (steps 5, 6 and Figure 3c).



Pipeline 1: DREAM.3D RATS Classification

An unsigned 8-bit integer array was created from the RATS mask (steps 7 and 8) and the holes inside the RATS mask are filled using the *ITK::Grayscale Fillhole Image Filter* to create a solid mask of the part (step 9 and Figure 3d). Pixels in the Image and Gradient Magnitude arrays outside the part mask were set to zero using the 8-bit integer array with the *ITK::Mask Image Filter* (steps 10 and 11).

The next group of filters created another 8-bit integer array from the RATS mask but the inner sections, rather than being filled, were set to 1's with all other pixels set to 0's which created the classified array for this SEM image. (steps 12-18 and Figure 3e). The final step exported the data array in DREAM.3D format. As expected, the images show the RATS threshold boundaries appear where the gradient magnitude is strongest.

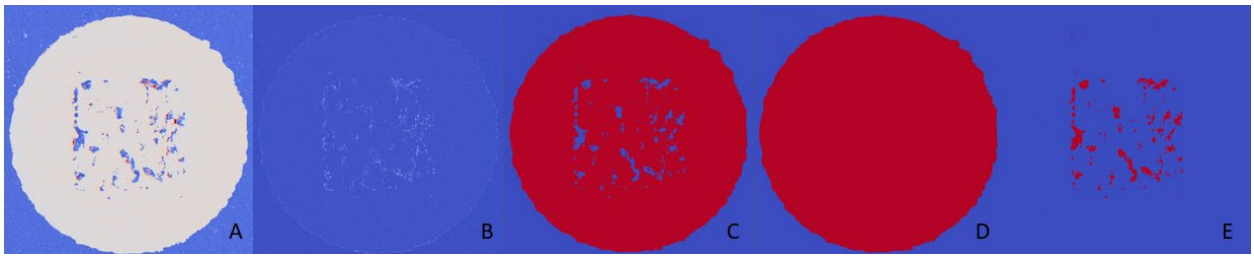


Figure 3 RATS classification of SEM TLD BSE 0º SEM Image: Original SEM image read into DREAM.3D (a), Gradient Magnitude (b), RATS (c), Part Mask (d), RATS Classification (e)

3.1.2 XCT with RATS

The RATS algorithm was not an effective classification technique for the XCT array. Following the same basic pipeline as the SEM images shown in section 3.1.1, the 3D array was imported into DREAM.3D, a Gradient Magnitude array was created, and the RATS algorithm was implemented. Figure 4 shows an example layer, approximately in the center of the part, at slice 853, normal to the Z-axis. This part was printed with an extra boundary scan which is particularly dense. Due to this boundary scan, the magnitude of the density gradient at the boundary is significantly higher than the gradients inside the part that indicate if a void may be present. This caused the RATS algorithm to threshold the 3D image based on the high magnitude of gradient at the boundary and does not threshold along the lower interior gradients.

Comparing Figure 3 for the SEM image and Figure 4 for the XCT slice provides a visualization of this effect with the Gradient Magnitude of each image is shown in (b) and initial RATS classification shown in (c). For the SEM images, the gradient is of similar magnitude at both the exterior boundary of the part and at the interior features which resulted in the interior features being selected by RATS. For the slice of the XCT array, the gradient at the exterior of the part is of a much higher magnitude than the gradient at the features in the interior which resulted in the interior features being excluded by RATS.

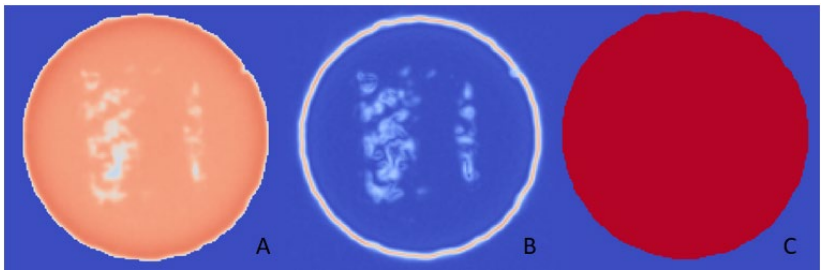


Figure 4 RATS classification of XCT array at Z-853: Original density values read into DREAM.3D (a), Gradient Magnitude (b), RATS (c)

3.2 Otsu Classification

The objective of Otsu's method is to minimize within-class variance by searching for a global optimal threshold and computing a gray-level histogram. Otsu's method [19] searches for a threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_o(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \quad [5]$$

The pixels of a given image are represented in L gray levels. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \ldots + n_L$. The gray-level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N}, \qquad p_i \geq 0, \sum_{i=1}^{L} p_i = 1 \quad [6]$$

Pixels are segregated into two classes, C_0 and C_1, using a threshold at level k. The probabilities of class occurrence and the class mean levels, respectively, are given by:

$$\omega_o(t) = \sum_{i=0}^{t-1} p(i), \qquad \omega_o(t) = \sum_{i=0}^{t-1} p(i) \quad [7, 8]$$

$$\mu_o(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_o(t)}, \qquad \mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)} \quad [9, 10]$$

The total mean-level of the original picture is:

$$\mu_T = \sum_{i=0}^{L-1} ip(i) \quad [11]$$

Otsu's method chooses the optimal threshold by maximizing the between-class variance, which is equivalent to minimizing the within-class variance. For two classes:

$$\sigma_b^2(t) = \sigma^2 - \sigma_\omega^2(t) = \omega_o(\mu_o - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_o(t)\omega_1(t)[\mu_o(t) - \mu_1(t)]^2 \quad [12]$$

The procedure of Otsu's method for threshold selection is:

1) Compute histograms and probabilities of each intensity level
2) Set up initial $\omega_i(0)$ and $\mu_i(0)$
3) Step through all possible thresholds from t =1 to maximum intensity
   a. Update ω_i and μ_i
   b. Compute $\sigma_b^2(t)$
4) Find threshold that corresponds to maximum $\sigma_b^2(t)$

Thresholding using Otsu's method is implemented on image-type datasets from the ITK library with the DREAM.3D filter *ITK::Otsu Multiple Thresholds Image Filter*.

3.2.1 SEM with Otsu

Each of the 8 SEM imaging positions were classified using Otsu's method. An example pipeline is shown on the left. Prior to classification, the data were preprocessed in DREAM.3D with similar filters as the RATS method outlined in section 3.1.1 (through step 11). Otsu thresholding, with one threshold to create a binary array, is performed in two ways.

Pipeline	
01	ITK::Image Reader
02	Set Origin & Spacing (Image)
03	ITK::Median Image Filter
04	ITK::Gradient Magnitude Image Filter
05	Robust Automatic Threshold
06	Isolate Largest Feature (Identify Sample)
07	Create Data Array
08	Replace Value in Array (Conditional)
09	ITK::Grayscale Fillhole Image Filter
10	ITK::Mask Image Filter
11	ITK::Mask Image Filter
12	ITK::Otsu Multiple Thresholds Image Filter
13	Replace Value in Array
14	Replace Value in Array
15	Replace Value in Array
16	ITK::Mask Image Filter
17	ITK::Grayscale Fillhole Image Filter
18	ITK::Grayscale Fillhole Image Filter
19	ITK::Median Image Filter
20	ITK::Otsu Multiple Thresholds Image Filter
21	Write DREAM.3D Data File

Pipeline 2: DREAM.3D Otsu Classification for SEM

For the first method, the *ITK::Otsu Multiple Thresholds Image Filter* was applied to the SEM Image (step 12, Figure 5a) to create a binary image. As with the RATS technique, a classified array for this SEM image was created by setting the inner sections to 1's with all other pixels set to 0's (steps 12-16). Finally, the *ITK::Grayscale Fillhole Image Filter* was used to fill the holes in the resulting array (step 17, Figure 5b).

The second method applied the *ITK:Grayscale Fillhole Image Filter* to the Gradient Magnitude array (step 18, Figure 5c) to create an array with the gradient values filled where the gradients were continuous. An array was then created using the median values, at r=2, of that Fillhole array (step 19, Figure 5d). Finally, the Otsu Threshold was found and implemented on the final Median Fillhole Gradient Magnitude array to create the second binary classified array using Otsu's method (step 20, Figure 5e) and the array was exported in DREAM.3D format.

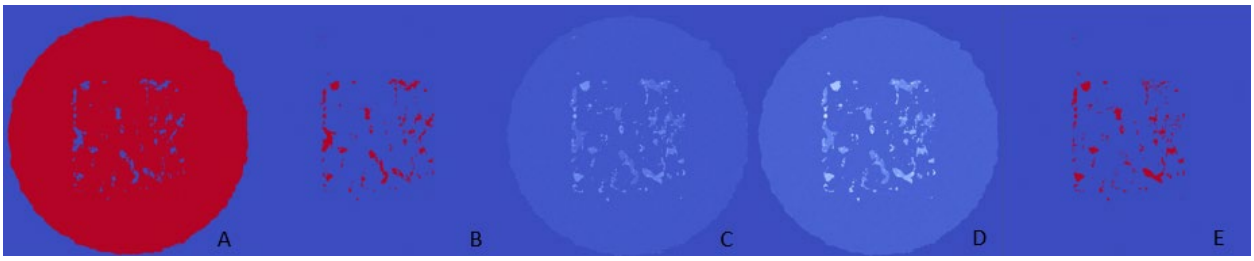


Figure 5 Binary Otsu classification of SEM TLD BSE 0º SEM Image: Otsu's method applied to the SEM image (a and b), Gradient Magnitude Fillhole (c), Median Gradient Magnitude Fillhole (d), Otsu's method applied to the Median Gradient Magnitude Fillhole (e)

The previous implementation of Otsu's method determined one threshold which created binary classes (0's and 1's). The DREAM.3D implementation of Otsu's method can easily create multivariate classifications of image arrays by adjusting the NumberOfThresholds input. As an example, Figure 6 shows the classification of the SEM image and the Median Gradient Magnitude Fillhole array for three and five classes. By removing classes 1 and 2 in the five-class execution of the Median Gradient Magnitude Fillhole array, four classes remain which can be used to demonstrated higher risk of a defect in that region (Figure 6e).

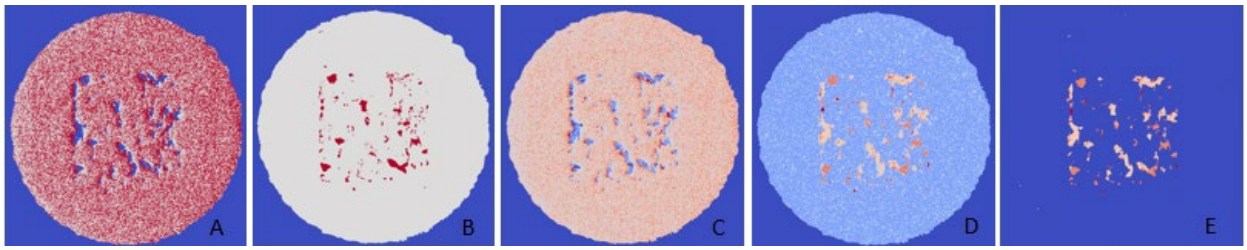
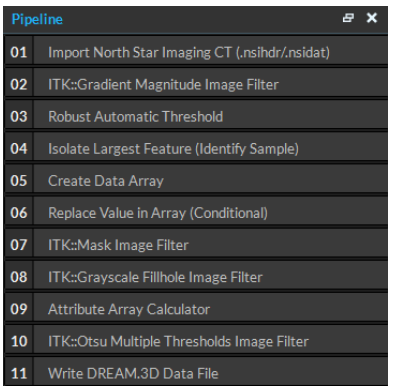


Figure 6: Multivariate Otsu classification of SEM TLD BSE 0º SEM Image: Three and five classes of the SEM image (a and b), three and five classes of the Median Gradient Magnitude Fillhole array (c and d), classes 1 and 2 removed from image d (e).

3.2.2 XCT with Otsu

Otsu thresholding of the 3D XCT array followed the basic setup procedure of the 2D SEM images discussed in section 3.2.1. The full pipeline is shown on the left. Importing with ITK:Image Reader and Set Origin & Spacing (Image) is not necessary for the XCT array as the NSI header file included voxel dimensions and other features of the dataset. Figures 7a and 7b shows the initial XCT array sliced normal to the Z and X planes. The Gradient Magnitude was calculated from the 3D array and RATS was implemented and *Isolate Largest Feature* defined the boundary of the part (steps 2-4). An unsigned 8-bit integer array was created as a mask of the part (steps 5-7). The *ITK::Grayscale Fillhole Image Filter* was applied to the original XCT array to create a new array with the pores in the XCT array filled (step 8). The *Attribute Array Calculator* filter was used to subtract the Fillhole array from step 8 from the XCT array which create an array of Inverse Fillhole leaving only the pores from the original XCT array (Figure 7 (c-d) and step 9). Finally, Otsu's threshold was implemented on the Inverse Fillhole array to provide a final classification of the XCT array in 3D (Figure 7 (e-f) and step 10). Figure 7g shows a 3D representation of the classified defects for this XCT dataset. Defect regions are clearly visible in the lettering sections printed at lower power and this part has more defects in the +Z region which is later in the build. This Inverse Fillhole procedure is equivalent to the procedure to build the Otsu Image array in section 3.2.1 and shown in Figure 7 (a-b).



Pipeline 3: DREAM.3D Otsu Classification for XCT

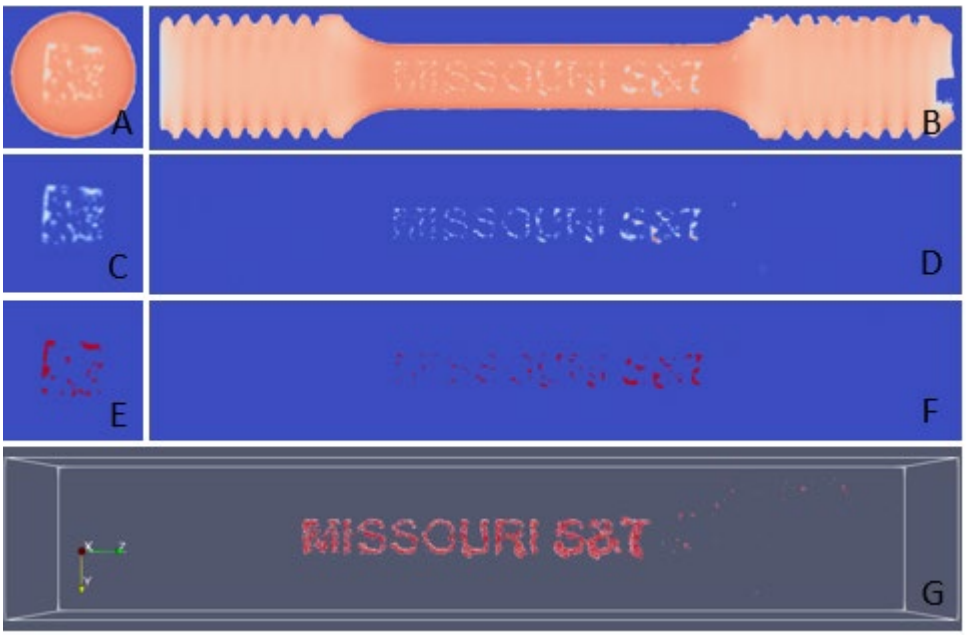


Figure 7 XCT sliced on Z-522 (a, c, e) and X-140 (b, d, f): Initial XCT array (a, b), Inverse Fillhole (c, d), Classified by Otsu's method (e, f), 3D visualization of classified defects (g)

As with the SEM images in section 3.2.1, the 3D XCT array can be classified using Otsu's method with a different number of thresholds. Figure 8 demonstrates the use of three thresholds to create four classes (a and b) and five thresholds to create six classes (b and c). As with the previous figure, slice 522 normal to the Z axis is shown (a and c). This method of classification may be used to show higher risk regions of defect occurrence in a predictive model.

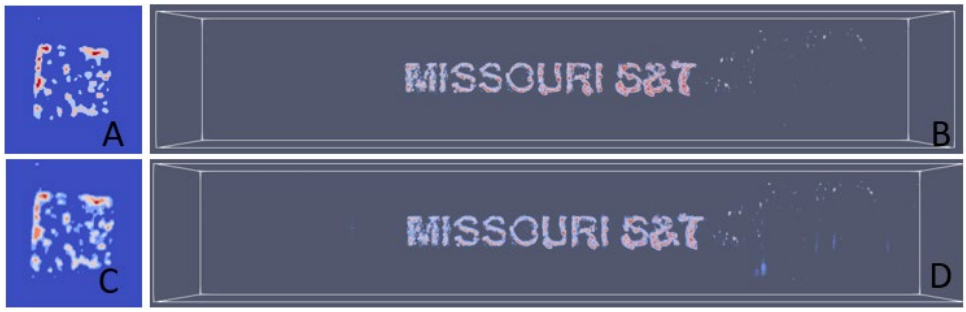


Figure 8 XCT with multivariate classifications: Four classes (a and b) and Six classes (c and d)

3.3 K-means Classification

K-means is a local optimal method that differs from Otsu's method because it does not compute a gray-level histogram. K-means is thought to be more efficient for multilevel thresholding and multidimension datasets [20]. In K-means [21], the objective is to minimize the sum-of-squares criterion (within-class variance):

$$J = \sum_{j=1}^{k} \sum_{n \in C_j} \left(x_n - U_j\right)^2 \quad [13]$$

Where N data points are partitioned into k disjointed subsets $\{C_1, C_2, \ldots, C_k\}$. Additionally, x_n is a vector representing the n^{th} data point, j is the index of classes and $1 < j < k$, U_j is the centroid of the data points in C_j, and C_j contains N_j data points.

The procedure of K-means for threshold selection is:

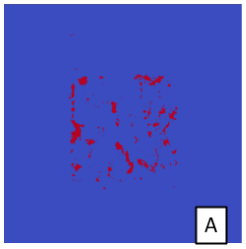
1) Select k points as the initial class centroids
2) Assign each object to the class whose center it is closest to
3) When all objects have been assigned, recalculate the positions of the k centroids
4) Repeat step 2 and 3 until the positions of centroids no longer change
5) Find thresholds from the final partition

Thresholding using K-means is implemented on image-type datasets with DREAM.3D filter *K Means (ImageProcessing)*.

3.3.1 SEM with K-means

K-means classification was implemented for each of the 8 SEM images. The process is nearly identical to the pipeline in section 3.2.1 for SET classification with Otsu's method except the *K Means (ImageProcessing)* filter is used in place of the *ITK::Otsu Multiple Thresholds Image Filter* (step 12). Unlike the pipeline for Otsu's method, K-means classification is only performed on the SEM image itself and the method was not applied to a Median Gradient Magnitude Fillhole array because that array type is not an unsigned 8-bit integer and the DREAM.3D implementation of K-means requires an unsigned 8-bit integer.

Classification by K-means of the SEM images was found to be nearly identical to classification by Otsu's method on the same SEM images. For 0º SEM Imaging Position, Otsu's method classified 243,551 pixels in class 1 and K-means classified 244,405 pixels in that class. Figure 9a for K-means can be compared to Figure 7e for a visualization of both classification methods. Figure 9b is the confusion matrix of Otsu's method vs K-means showing that there were zero pixels in class 1 by Otsu's method that K-means did not segregate into class 1. As a matter of real performance, this is only a 0.3% difference in the number of pixels in class 1.



12,005,595	0
854	243,551

B

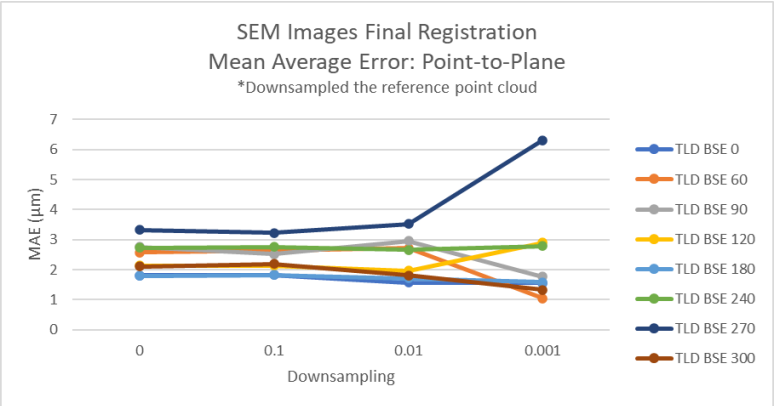
3.3.2 XCT with K-means

K-means classification was not applied to the XCT array because DREAM.3D requires the array that needs to be clustered to be an unsigned 8-bit integer. Following the pipeline in 3.2.2, except replacing the *K Means (ImageProcessing)* filter is in place of *the ITK::Otsu Multiple Thresholds Image Filter*, the array for K-means to classify is the Inverse Fillhole which was unable to be converted to the required format. The inability to use K-means on the 3D XCT array is the same reason that K-means could not be used to classify the Median Gradient Magnitude Fillhole SEM array in section 3.3.1.

4.0 Improving SEM Classification by Fusing Multiple Imaging Positions

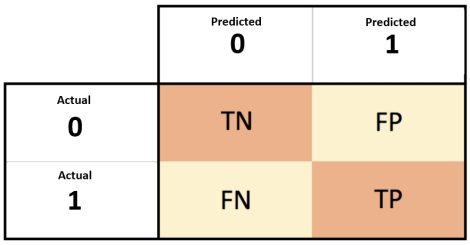
This section outlines the method used to measure performance of each of the four binary SEM classification techniques (RATS, Otsu of the SEM Images, Otsu of the Median Gradient Magnitude Fillhole array, and K-means of the SEM Images). The four classification techniques were combined in a boosted array that improved classification for each SEM Image. Finally, the eight boosted SEM Imaging Positions are combined to improve the overall classification.

A human labeled image was created that manually classified pixels to serve as the "ground truth" for this study. Each of the eight SEM Imaging Positions were transformed to the labeled image with an average registration accuracy of 2.4 μm, ranging from 1.8 μm for the closest registered Imaging Positions (0º and 180º) and 3.3 μm for the furthest registered Imaging Position (270º) [22]. The length and width of each pixel in the images are 1.25 μm. Figure 10 shows the Point-to-plane Mean Average Error (MAE) calculated for each of the eight SEM Imaging Positions.



4.1 Classification Metric

A classification metric was utilized that scores each binary classification array vs the human labeled ground truth image. This metric includes a 4x4 confusion matrix counting pixels in the classified SEM image (0 or 1) that are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).



The classification metric found Precision, Recall, F1-score, Accuracy, and Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) score. Classification performance can be shown to be very high on individual measures but different measures are used to show performance on all aspects of the classifier rather than maximizing one feature. Each measure in the classification metric is shown in equations 8 to 12 below.

$$Recall = \frac{TP}{TP+FN} \quad [14]$$

$$Precision = \frac{TP}{TP+FP} \quad [15]$$

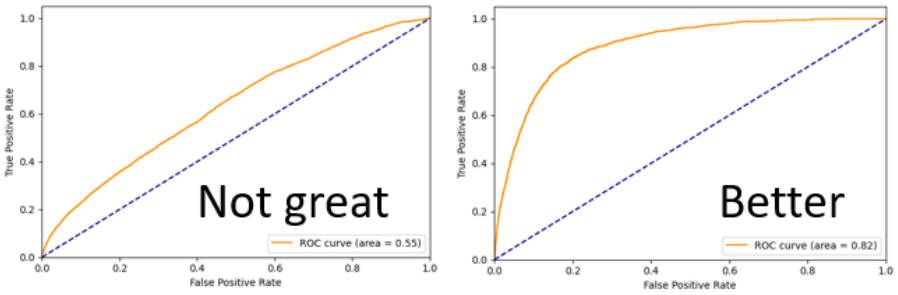
$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad [16]$$

$$Accuracy = \frac{TN+TP}{TN+FP+FN+TP} \quad [17]$$

$$ROC - AUC = \int_{x=0}^{1} TPR(FPR^{-1}(x)dx \quad [18]$$

Recall is a classifier's ability to find all the positive samples, Precision is the ability to not label a sample positive if it is negative, and F1-score measures recall and precision at the same time. Accuracy is simply the percentage of True values in the set.

ROC-AUC represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. True Positive Rate (TPR) and False Positive Rate (FPR) are plotted (orange lines in Figure 12) and ROC-AUC score is found by calculating the area under the curve. A perfect performing classifier will have a ROC-AUC score of 1 with decreasing performance lower than 1. A ROC-AUC score of 0.5 represents the equivalent of a random guess with 50% probability of success (blue dotted lines in Figure 12).



The classification metric was implemented in Python using the scikit-learn metrics module. Specific libraries that were used: confusion_matrix, classification_report, roc_auc_score, and roc_curve.

4.2 Individual SEM Classification Performance

Performance on each of the individual binary SEM images was measured with the classification metric from section 4.1. Table 1 shows the results for each of the four binary TLD BSE 0º classified arrays.

Table 1: Classification Report for the four classified TLD BSE 0º arrays

		Confusion Matrix		Classification Report						
Image	**Array**			class	**precision**	**recall**	**f1-score**	**test support**	**accuracy**	**AUC**
TLD BSE 0	RATS	11,877,364	26,423	0	0.99	1	1	11,903,787	0.99	0.88
		82,106	264,107	1	0.91	0.76	0.83	346,213		
TLD BSE 0	Otsu Image	11,895,845	7,942	0	0.99	1	1	11,903,787	0.99	0.84
		110,604	235,609	1	0.97	0.68	0.8	346,213		
TLD BSE 0	Otsu M2 FH GM	11,874,505	29,282	0	0.99	1	1	11,903,787	0.99	0.80
		134,604	211,609	1	0.88	0.61	0.72	346,213		
TLD BSE 0	K means	11,895,555	8,232	0	0.99	1	1	11,903,787	0.99	0.84
		110,274	235,939	1	0.97	0.68	0.8	346,213		

For the TLD BSE 0º SEM image, the performance of each of the four classifiers are fairly close to each other with the Otsu M2 FH GM classifier being the lowest performance (class 1 F1 of 0.72 and AUC of 0.80) and the RATS classifier scoring the highest (class 1 F1 of 0.83 and AUC of 0.88). As discussed in section 3.3.1, Otsu Image and K means resulted in very similar

classification and scored very close to each other, both having the same class 1 F-1 and AUC scores of 0.80 and 0.84, respectively.

Overall performance was improved by combining each of the four classifiers for the individual SEM Imaging Positions. To create a combined array, a pixel was assigned to Class 1 if *any* of the four classifiers assigned it to Class 1. Figure 13 shows the classified array for the TLD BSE 0º image with each of the four classified arrays (a-d) and the combined overall classified array for this image (e).

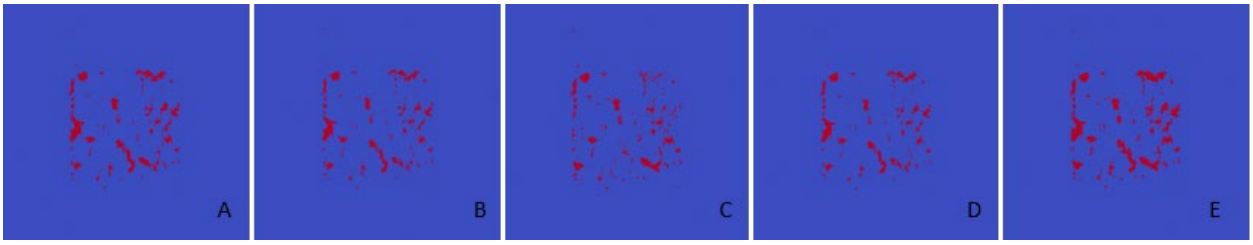


Figure 13 TLD BSE 0º Classification: RATS (a), Otsu Image (b), Otsu M2 FH GM (c), K means (d), Combined classification (e)

This combining operation is performed in DREAM.3D using the *Attribute Array Calculator* filter which allows mathematical calculations on arrays. Each of the four classified arrays are added together and all pixels with values greater than 1 are set to 1 using the *ITK::Binary Threshold Image Filter*.

Table 2 shows the result of combining the four classifiers for each of the 8 SEM Imaging Positions. Performance on the TLD BSE 0º image was improved by 5 points on the Class 1 F1-score, up to 0.88, and by 7 points on the AUC score, up to 0.95. The best classification performance was on the TLD BSE 180º image (0.9 Class 1 F1-score and 0.96 AUC) and the worst classification performance was on the TLD BSE 90º image (0.86 Class 1 F1-score and 0.94 AUC.

Table 2: Classification report for the combined classifications of each SEM Imaging Position

Image	Confusion Matrix		Classification Report						
			class	precision	recall	f1-score	test support	accuracy	AUC
TLD BSE 0	11,853,098	50,796	0	1	1	1	11,903,894	0.99	0.95
	32,612	313,494	1	0.86	0.91	0.88	346,106		
TLD BSE 60	11,848,479	55,415	0	1	1	1	11,903,894	0.99	0.94
	38,487	307,619	1	0.85	0.89	0.87	346,106		
TLD BSE 90	11,844,676	59,218	0	1	1	1	11,903,894	0.99	0.94
	42,523	303,583	1	0.84	0.88	0.86	346,106		
TLD BSE 120	11,849,197	54,697	0	1	1	1	11,903,894	0.99	0.95
	35,116	310,990	1	0.85	0.9	0.87	346,106		
TLD BSE 180	11,857,014	46,880	0	1	1	1	11,903,894	0.99	0.96
	24,293	321,813	1	0.87	0.93	0.9	346,106		
TLD BSE 240	11,844,407	59,487	0	1	1	1	11,903,894	0.99	0.95
	30,647	315,459	1	0.84	0.91	0.87	346,106		
TLD BSE 270	11,841,487	62,407	0	1	0.99	1	11,903,894	0.99	0.94
	36,528	309,578	1	0.83	0.89	0.86	346,106		
TLD BSE 300	11,840,638	63,256	0	1	0.99	1	11,903,894	0.99	0.95
	35,509	310,597	1	0.83	0.9	0.86	346,106		

4.3 Fusion of SEM Imaging Positions

The eight SEM Imaging Positions can be fused together by selecting a threshold where a pixel is segregated into Class 1 if it is labeled Class 1 in *n* number of Imaging Positions greater than or equal to the threshold. Section 4.2 followed the same fusion process with *n*>=1 where a pixel was assigned to class 1 if any classifier assigned it to Class 1.

Table 3 shows the classification report of varying the selection threshold from 1 to 8 where n>=1 classifies a pixel as Class 1 if it is labeled Class 1 in any classified array and n=8 where a pixel is classified as Class 1 if all eight arrays label it a Class 1. Figure 14 plots the precision, recall, and F1-score on Class 1 as well as the ROC-AUC for the overall classifier for each threshold.

Table 3: Classification report for the combined classification of all eight SEM Imaging Position where *n* is the number of Imaging Positions a pixel must be in Class 1 to be labeled Class 1.

Image	Array	Confusion Matrix		Classification Report						
				class	precision	recall	f1-score	test support	accuracy	AUC
Combined	n>=1	11,798,268	105,626	0	1	0.99	0.99	11,903,894	0.99	0.99
		5,577	340,529	1	0.76	0.98	0.86	346,106		
Combined	n>=2	11,827,601	76,293	0	1	1	1	11,903,894	0.99	0.98
		8,468	337,638	1	0.82	0.98	0.89	346,106		
Combined	n>=3	11,847,196	56,698	0	1	1	1	11,903,894	0.99	0.98
		14,767	331,339	1	0.85	0.96	0.9	346,106		
Combined	n>=4	11,858,645	45,249	0	1	1	1	11,903,894	0.99	0.97
		22,163	323,943	1	0.88	0.94	0.91	346,106		
Combined	n>=5	11,869,969	33,925	0	1	1	1	11,903,894	0.99	0.95
		32,415	313,691	1	0.9	0.91	0.9	346,106		
Combined	n>=6	11,879,989	23,905	0	1	1	1	11,903,894	0.99	0.93
		46,354	299,752	1	0.93	0.87	0.9	346,106		
Combined	n>=7	11,889,403	14,491	0	0.99	1	1	11,903,894	0.99	0.90
		66,645	279,461	1	0.95	0.81	0.87	346,106		
Combined	n=8	11,895,738	8,156	0	0.99	1	1	11,903,894	0.99	0.86
		98,771	247,335	1	0.97	0.71	0.82	346,106		

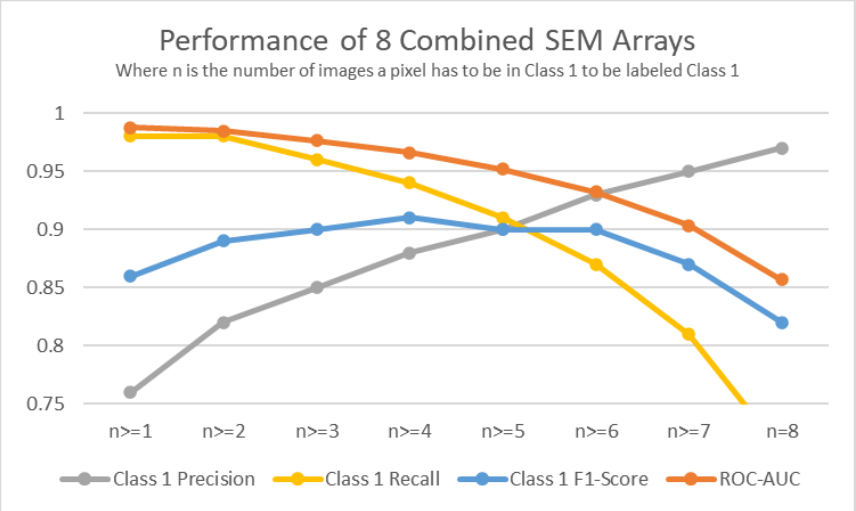


Figure 14: Classification performance was further improved by fusing multiple imaging positions.

The best measure for this instance is the Class 1 F1-Score because it solely focuses on performance of the defect class. Due to the highly unbalanced characteristics of this dataset, ROC-AUC, macro classifier scores that take into account both classes, and performance on the nominal class (class 0) can be very high if there is a small number of pixels labeled as a defect (that scenario scores very low on Class F1-Score). The optimal performance, based on Class 1 F1-Score, is found at n>=4 meaning a pixel was classified as a defect if it was in Class 1 in four or more of the 8 classified arrays. The optimal performance of the fused classifier resulted in a Class 1 F1-Score of 0.91 and ROC-AUC of 0.97, an improvement of the best individually classified SEM Imaging Position (TLD BSE 180º) by 1 point each.

The chart in Figure 14 shows that for low thresholds, where a pixel can be classified as a defect based on a smaller number of individual images, the precision is very low, and the recall is very high. This low threshold results means that the classifier is over predicting defects; it tends to find most of the defects but has more false positives. For high thresholds, where more images are needed to classify a pixel as a defect, the inverse is true, and the classifier tends to under predict defects. The results from the higher thresholds show very low recall and high precision. This means more defects are missed but there are less false positives.

5.0 Classification Verification

The classified XCT array developed in section 3.2 provided a method to develop ground truth labels which may be used to train a supervised classifier to predict the probability of defects. Otsu's method, classifying defect labels of the 3D XCT array, is unsupervised; this section outlines a method to verify the method's accuracy using the 2D SEM imaging. The approximate location of the extracted section where the SEM images were taken was known but there is not a way to tell exactly how much material was removed after the wire EDM, grinding, and polishing process described in Section 2.1. To verify the classification of the XCT array the 2D image is manually located in the 3D array in Section 5.1, an algorithm to search through the 3D array for a best-fit to the 2D image is developed in Section 5.2, and the accuracy of the 3D labeling compared to the 2D image is discussed in Section 5.3.

5.1 Manually Locate 2D Section in 3D XCT Array

Paraview was used to visualize layers normal to the z-axis in the XCT array at the approximate region of the extracted 2D section. Recall from Section 2.2 that the layer height of the XCT images was 29.96 μm. Figure 15 shows consecutive layers in the XCT array which were compared with the SEM image.

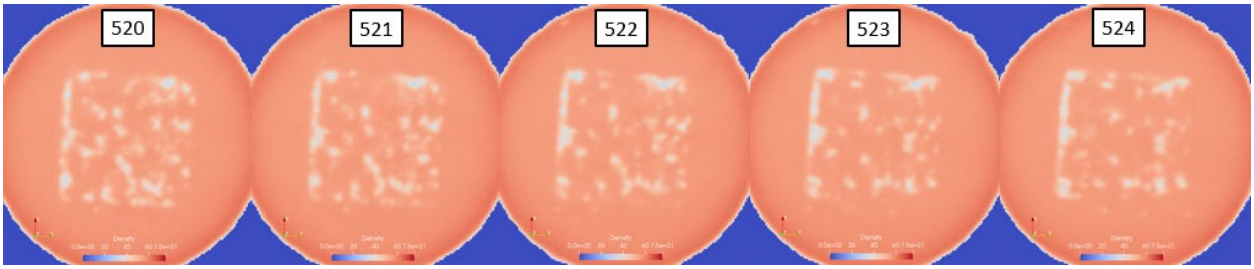


Figure 15: Consecutive layers on the XCT array near the location of the 2D extracted section

The labeled SEM image is shown in Figure 16. Based on the appearance of defects in the SEM image, the extracted section appears to be closest to the z-normal layer 521 in the XCT array.

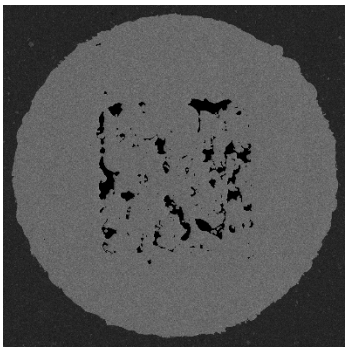


Figure 16: The human-labeled SEM image of the extracted section in the test piece

5.2 Automatically Locate 2D Section in 3D XCT Array

The manual location of the extracted section requires an analyst to interrogate layers of the XCT array and make a subjective decision as to which images match. In this section, a process was developed to automatically locate the extracted section using the human-labeled defect image generated from SEM. This provides a method to determine the location of a sampled section, accurate to the height of the XCT layers, after performing wire EDM, grinding, and polishing on a surface extracted from the test piece.

A surface point cloud of the SEM image was registered to the XCT array in DREAM.3D using Iterative Closest Point (ICP) [22]. The *Apply Transformation to Geometry* filter does not currently function on image-type geometries so the images were manually rotated, and the *Set Origin and Spacing (Image)* filter was used to apply the translation components of the Affine Transformation calculated by ICP. This manual process introduced error to the registration and the subsequent testing with the SEM image to the XCT array. Finally, the SEM image was down sampled to the lower resolution XCT grid using the *Fuse Regular Grid (Nearest Point)* filter which uses nearest-neighbor interpolation. Figure 17 shows the SEM point cloud registered to the XCT array (a), the z-522 layer in the XCT array (b), the labeled SEM image after rotation and translation (c), the labeled SEM image after down sampling to XCT resolution (d), and the labeled SEM image in the final two-class configuration (e).

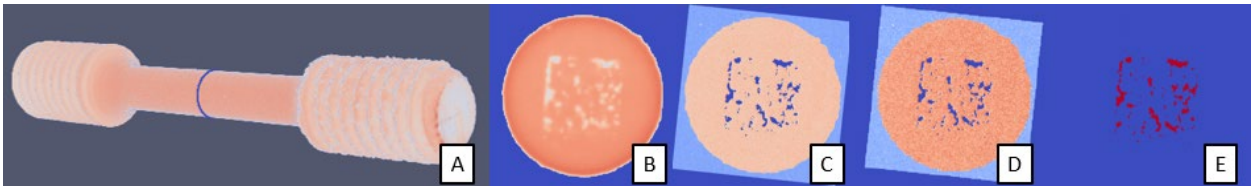


Figure 17 Registering and down sampling SEM to XCT

After the labeled SEM image was registered and down sampled to the XCT grid, a Python code was written to loop through each z-layer in the XCT array and measure the classification performance using the labeled SEM image as the ground truth array. As in previous sections, the Class 1 F1-Score and ROC-AUC performance were the best indicators of a well-matched layer.

Figure 18 shows the results of each z-layer in the XCT array (a) and a zoomed in section of the plot at the optimal values of F1 and ROC-AUC (b). The layers that were printed at lower power, roughly z-layer 500 to 1200, are seen in Figure 18a where the ROC-AUC and F1 scores are higher. A smaller number of defects occurred in the nominally printed area just after the low-power section and are seen roughly around z-layer 1250.

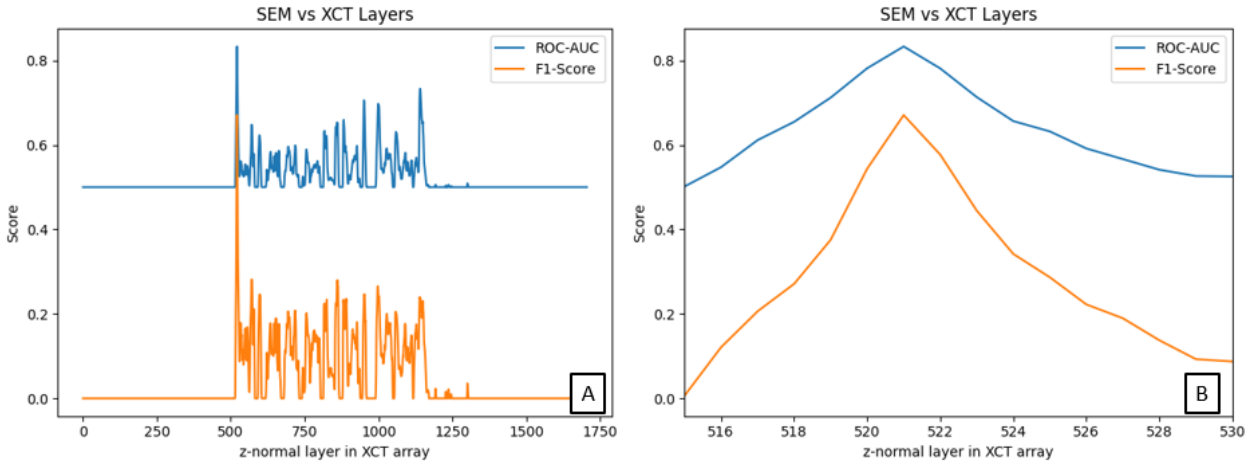


Figure 18 Plots of ROC-AUC and F1-Score for layers in the XCT compared to the human-labeled SEM image over the whole part (a) and zoomed in near the optimal value (b)

The best matched XCT layer, where F1 and ROC-AUC were maximum, was z-521. This layer was also the best subjective match from Section 5.1. Table 4 shows the full metrics report for layer z-521 and its closest neighbors. In all measures, including both precision and recall as well as accuracy, layer Z-521 outperformed the other layers and was shown to be the best match in the XCT array with the 2D image.

Table 4: Metrics report for layers in the XCT compared to the human-labeled SEM image.

	Confusion Matrix		Classification Report						
Z Layer in XCT			class	precision	recall	f1-score	test support	accuracy	AUC
Z-520	80,860	311	0	1	1	1	81,171	0.99	0.78
	260	340	1	0.52	0.57	0.54	600		
Z-521	80,976	195	0	1	1	1	81,171	1	0.83
	199	401	1	0.67	0.67	0.67	600		
Z-522	80,935	236	0	1	1	1	81,171	0.99	0.78
	261	339	1	0.59	0.56	0.58	600		

A second sample layer (Layer 2) was cut and polished, removing 37 µm from the first layer (Layer 1). SEM imaging was taken of this layer using the processed outlined in Section 2.1 capturing the image only at the 0º imaging position. Four individual classification arrays were created for this image using the techniques in Section 3.0 and a combined classification array was created using the technique in Section 4.2. A human-labeled ground-truth array for this sample layer was not created.

Following the same process as Layer 1, the classified SEM image for Layer 2 was registered to the XCT array and down sampled to the XCT grid. The Python code, measuring Class 1 F1-Score and ROC-AUC performance, was executed to show that the best matched layer in the XCT array is Z-522. This result is subjectively verified by comparing the SEM image to layer Z-522 shown in Figure 15. Figure 19 shows the SEM image taken of Layer 2, the combined classification array, and the F1-Score/ROC-AUC performance at neighboring layers to Z-522.

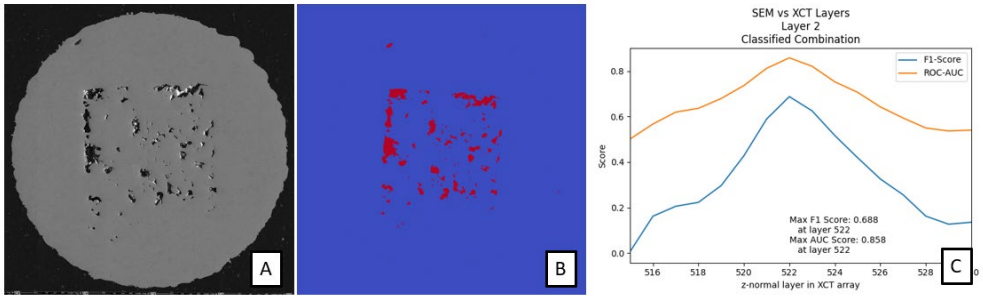


Figure 19 Layer 2 SEM image (a), combined classification results (b), and performance measures showing Z-522 is the matched layer in XCT array (c)

5.3 Classification Results

The peak of the F1 and ROC-AUC scores in section 5.2 demonstrated that sampled layer where SEM images of Layer 1 were taken lies at layer Z-521 +/- one layer and Layer 2 lies at layer Z-522 +/- one layer in the XCT array. The registration of the SEM array to the XCT array included manual steps which would introduce error in the resulting classification scores. A second source of error is due to the interpolation necessary when down sampling. With both error sources, the precision and recall on the defect class were similar, showing a balanced ability to find positive samples and not label a sample positive if it is negative.

For the 3D XCT array to be used for ground truth with in-situ data as inputs for a machine learning model, the XCT array must be fused with the in-situ data. Previous work has registered in-situ data from SWIR camera, with a pixel resolution of 125 µm x 125 µm in X and Y, to the CT array used in this study [22]. Both arrays, the human labeled SEM image and the XCT array were down sampled to the SWIR pixel resolution using the *Fuse Regular Grid (Nearest Point)* like in the SEM-to-XCT interpolation done in Section 5.2. The result of this down sampling operation for both arrays from Layer 1 are shown in Figure 20.

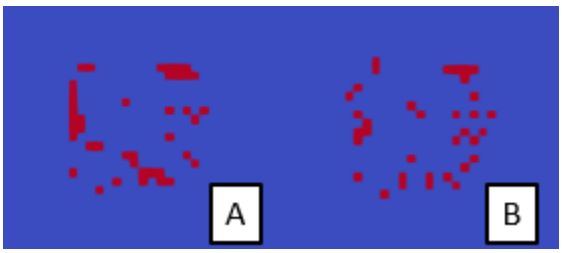


Figure 20: Matched XCT layer (a) and human-labeled SEM image of Layer 1 (b) after down sampling to SWIR camera resolution on layer Z-521

Recall that the original SEM pixel size was 1.25 µm x 1.25 µm and the XCT X-Y pixel size was 29.96 µm x 29.96 µm. The classified arrays of both layers were down sampled to the resolution of the SWIR camera of 125 µm x 125 µm. The method outlined in section 5.2 was followed to match the SEM images to layers in the SWIR array. After down sampling, Layer 1, which was Z-521 in the XCT array, is located at Z-303 in the SWIR array. Layer 2, which was at Z-522 in the XCT array, is located between Z-303 and Z-304 in the SWIR array. Figure 21 shows the matching performance on both layers in the SWIR array. Layer 2 is known to be 37 µm further into the part from Layer 1 and, with a Z-height of 50 µm in the SWIR array, it was expected that the sampled Layer 2 falls between SWIR layers.

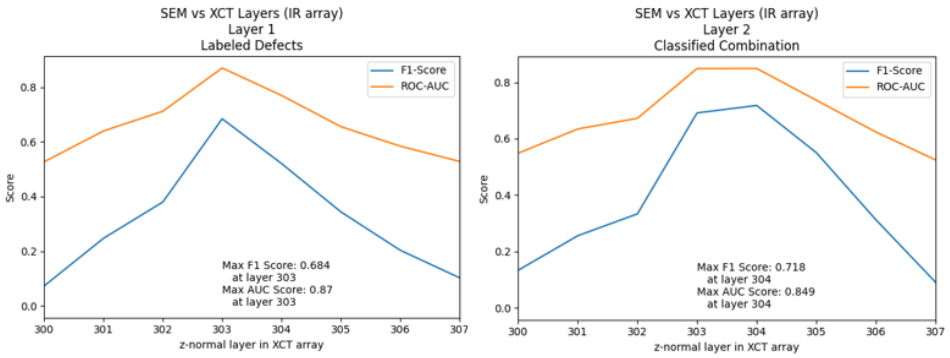


Figure 21: Two 2D sampled layers located in the 3D SWIR array.

The classification report scoring the XCT array at Z-303 against the Labeled SEM image for Layer 1 is shown in Table 5. The number of pixels in class 1 was reduced from 600 to 35 for the SEM image and from 575 to 41 for the XCT array. After introducing error due to registration and interpolation, the classification on the defect class is roughly the same with an F1-Score of 0.68.

Table 5: Metrics report for matched XCT layer compared to the human-labeled SEM image from Layer 1 after down sampling to SWIR camera resolution

	Confusion Matrix		Classification Report						
Z Layer in XCT			class	precision	recall	f1-score	test support	accuracy	AUC
Z-303	5,746	15	0	1	1	1	5,761	1	0.87
	9	26	1	0.63	0.74	0.68	35		

6.0 Discussion of Results and Conclusions

This study proposed a defect classification technique for a 3D array that can be used as ground truth labels for model training. Otsu's method, implemented with DREAM.3D, was shown as the most effective way to classify XCT data. The classification method was validated using SEM imaging of a sample layer and a method to locate the sample layer in a 3D XCT stack has

been developed. Future work can be done using ground truth labels from this study to build a predictive quality model based on in-situ data from LPBF builds.

K-means and Otsu's method were shown to have very similar results on the 2D images. DREAM.3D was not able to implement K-means classification on the 3D XCT array but DREAM.3D is able to quickly implement Otsu's method by leveraging the ITK library. Otsu's method is the preferred technique for classifying data, both in binary modes and multivariate modes, for this study. As of the time of writing, there is not a K-means implementation from the ITK toolbox implemented in DREAM.3D and the implementation of K-means runs noticeably slower than Otsu's method on the 2D SEM images even though the reviewed literature predicts K-means to run faster. Both the speed of the Otsu implementation and the ability to run on multiple data types highlights the power of leveraging the ITK toolbox for filters within DREAM.3D.

Both Otsu's method and K-means can be used for multivariate classification of both the SEM and XCT datasets. RATS creates local threshold lines and can only be used for binary classification. In addition to binary classification, the multivariate modes demonstrated with Otsu's method for both SEM (section 3.2.1) and XCT (section 3.2.2) demonstrate the potential to use this type of classification to show higher risk regions. This may be of particular interest for multivariate machine learning models and making risk assessments based on those model predictions.

Defect classification was shown to be improved by fusing multiple imaging positions. Fusing multiple imaging positions only resulted a 1-point gain in both Class 1 F1 and ROC-AUC scores over the highest performing individual image. However, when a human labeled "ground truth" array is not available there is not a way to tell which individual image is the highest performing classifier. The average individual Class 1 F1-score was 0.87 and the lowest was 0.86. Fusing the individual images is very simple using the developed DREAM.3D pipeline allows for quick overall gains in performance as well as the ability to prioritize precision or recall performance based on the specific application.

Accuracy of registration affects defect classification performance. Defect labeling of the SEM images can potentially be improved with a more accurate registration. DREAM.3D currently cannot apply the transformations that are computed internally. Further development of the *Apply Transformation to Geometry* filter to work with image-type datasets and apply interpolation would fully integrate registration and improve accuracy.

Further investigation with additional SEM layers and/or collecting serial sectioning would help further verify classification of the XCT defects. The process developed to automatically locate the 2D sample in the 3D array was proven to be effective. The location algorithm is based on internal features so it would not be possible to locate a sample if internal features were not present or detected. The ability to locate a sample taken may allow further investigation to use serial sectioning to "biopsy" specific areas of interest found in an XCT array.

7.0 Acknowledgements

This work was funded by the Center for Aerospace Manufacturing Technologies and The Boeing Company.

8.0 References

[1] ISO / ASTM52900-15, Standard Terminology for Additive Manufacturing – General Principles – Terminology, ASTM International, West Conshohocken, PA, 2015, www.astm.org

[2] M. Schmidt, M. Merklein, D. Bourell, D. Dimitrov, T. Hausotte, K. Wegener, L. Overmeyer, F. Vollertsen, G. Levy, Laser Based Additive Manufacturing in Industry and Academia, CIRP Annals, Volume 66, Issue 2, 2017, ISSN 0007-8506, https://doi.org/10.1016/j.cirp.2017.05.011.

[3] S. Everton, M. Hirsch, P. Stavroulakis, R. Leach, A. Clare, Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. 2016. Materials & Design. 95. 431-445. 10.1016/j.matdes.2016.01.099.

[4] A. Thompson, I. Maskery, R.K. Leach, X-ray Computed Tomography for Additive Manufacturing: A Review, Meas. Sci. Technol., 2016, 27(7), p 072001–072001. https://doi.org/10.1088/0957-0233/27/7/072001

[5] H. Carlton, A. Haboub, G. Gallegos, D. Parkinson, A. MacDowell, Damage Evolution and Failure Mechanisms in Additively Manufactured Stainless Steel, Materials Science and Engineering A 651, 2015, DOI 10.1016/j.msea.2015.10.073.

[6] V. Sundar, Z. Snow, J. Keist, G. Jones, R. Reed, E. Reutzel, Flaw Identification in Additively Manufactured Parts Using X-ray Computed Tomography and Destructive Serial Sectioning, Journal of Materials Engineering and Performance, 2021, https://doi.org/10.1007/s11665-021-05567-w

[7] J.B. Forien, N. Calta, P. DePond, G. Guss, T. Roehling, M. Matthews, Detecting keyhole pore defects and monitoring process signatures during laser powder bed fusion: A correlation between in situ pyrometry and ex situ X-ray radiography, Additive Manufacturing, Volume 35, 2020, 101336, ISSN 2214-8604, https://doi.org/10.1016/j.addma.2020.101336.

[8] J. Mitchell, T. Ivanoff, D. Dagel, J. Madison, B. Jared, Linking pyrometry to porosity in additively manufactured metals, Additive Manufacturing, Volume 31, 2020, 100946, ISSN 2214-8604, https://doi.org/10.1016/j.addma.2019.100946.

[9] G. Mohr, S.J. Altenburg, A. Ulbricht, P. Heinrich, D. Baum, C. Maierhofer, K. Hilgenberg, In-Situ Defect Detection in Laser Powder Bed Fusion by Using Thermography and Optical Tomography—Comparison to Computed Tomography. Metals 2020, 10, 103. https://doi.org/10.3390/met10010103

[10] J. Liu, B. Jalalahmadi, YB. Guo, M. Sealy, N. Bolander, Nathan, A review of computational modeling in powder-based additive manufacturing for metallic part qualification. Rapid Prototyping Journal. 2018. 24. 10.1108/RPJ-04-2017-0058.

[11] I. Raffeis, F. Adjei-Kyeremeh, U. Vroomen, E. Westhoff, S. Bremen, A. Hohoi, A. Bührig-Polaczek, Qualification of a Ni-Cu Alloy for the Laser Powder Bed Fusion Process (LPBF):

Its Microstructure and Mechanical Properties. 2020. Applied Sciences. 10. 10.3390/app10103401.

[12] C.S. Lough, X. Wang, C. Smith, R. Landers, D. Bristow, J. Drallmeier, B. Brown, E. Kinzel, Correlation of SWIR imaging with LPBF 304L stainless steel part properties, Additive Manufacturing, Volume 35, 2020, 101359, ISSN 2214-8604, https://doi.org/10.1016/j.addma.2020.101359.

[13] H. Johnson, M. McCormick, L. Ibanez. The ITK Software Guide: Design and Functionality. Fourth Edition.2015. ISBN: 9781-930934-28-3

[14] T.S. Yoo, M. J. Ackerman, W. E. Lorensen, W. Schroeder, V. Chalana, S. Aylward, D. Metaxas, R. Whitaker, Engineering and Algorithm Design for an Image Processing API: A Technical Report on ITK - The Insight Toolkit. Stud Health Technol Inform. 2002; 85:586-92. PMID: 15458157.

[15] M.A. Groeber, M.A. Jackson, DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D, Integrating Materials 3, 56–72 (2014). https://doi.org/10.1186/2193-9772-3-5

[16] A.S.T.M. E8/E8M, Standard Test Methods for Tension Testing of Metallic Materials, ASTM International, West Conshohocken, PA, 2016.

[17] M.H.F. Wilkinson, F. Schut, Digital Image Analysis of Microbes: Imaging, Morphometry, Fluorometry and Motility Techniques and Applications, 1998, ISBN: 0-471-97440-4

[18] R. Szeliski, Computer Vision: Algorithms and Applications, 2011, ISBN: 978-1-84882-934-3

[19] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.

[20] D. Liu, J, Yu, Otsu Method and K-means, 2009 Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 344-349, doi: 10.1109/HIS.2009.74.

[21] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, volume 1:281–296, 1967.

[22] A. Lang, C. O. Rios, J. Newkirk, R. Landers, J. Castle, D. Bristow, Image Registration and Matching Error in 2D and 3D for Laser Powder Bed Fusion, Solid Freeform Fabrication Symposium, 2021