

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/332048338>

# Defect detection using trainable segmentation

Conference Paper · March 2019

DOI: 10.1117/12.2521768

CITATIONS

6

READS

577

3 authors:



**Bisma Mutiargo**

Agency for Science, Technology and Research (A\*STAR)

9 PUBLICATIONS 64 CITATIONS

SEE PROFILE



**Amin Garbout**

Natural History Museum, London

41 PUBLICATIONS 565 CITATIONS

SEE PROFILE



**Andrew Alexander Malcolm**

Agency for Science, Technology and Research (A\*STAR)

51 PUBLICATIONS 489 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Advanced Post-Processing and Non-Destructive Evaluation for Additive Manufacturing [View project](#)



Soil Surface Matters [View project](#)

# Defect Detection using Trainable Segmentation

<sup>a</sup> Bisma Mutiargo\*, <sup>a</sup> Amin Garbout, <sup>a</sup> Andrew A. Malcolm  
<sup>a</sup> Intelligent Product Verification Group

Advanced Remanufacturing and Technology Centre (ARTC)

3 Cleantech Loop, #01-01, Singapore 637143

\*mutiargob@ARTC.a-star.edu.sg

## ABSTRACT

Additive manufacturing processes create the opportunity for freedom of design as it allows parts to be manufactured where conventional methods would fail. Printing methods such as selective laser sintering (SLM), electron beam melting (EBM) and others usually produce micro-porosity (Ng, Jarfors, Bi, & Zheng, 2009). These micro defects can have a major impact on the functionality and lifetime of the components.

X-ray Computed Tomography (XCT) is an image acquisition technique that allows a complete three-dimensional capture of an object including its internal features and structures. Typically, an XCT system captures many digital 2D radiography images as the sample is being rotated. A computed tomography algorithm then post-process the 2D images into a reconstructed 3D digital image that represents the scanned part. This technology is an established method of non-destructive evaluation (NDE) to detect the presence of cracks and large porosity in additively manufactured components.

However, micro - defects and cracks are known to be difficult to detect. The lack of distinction between X-ray artefacts due to scattering and beam hardening, and the defects make it impossible for simple intensity-based image processing algorithms, such as thresholding, to reliably detect and quantify the presence of a defect especially as the defect size approaches the imaging resolution.

Here, an approach to improve micro-porosity and crack detection through the use of random forest classifier was studied. This method was optimized to detect defects that are very close to the voxel size. To achieve this, trainable segmentation with a random forest classifier was used with three pre-defined classes (Pore, Material, and Air).

Random forest classifier is a general ensemble learning method that can be utilized for image classification. It creates a set of decision trees from randomly selected subsets of the training set. It then compiles the probability aggregate of each layer of the node within the tree to decide on the final outcome.

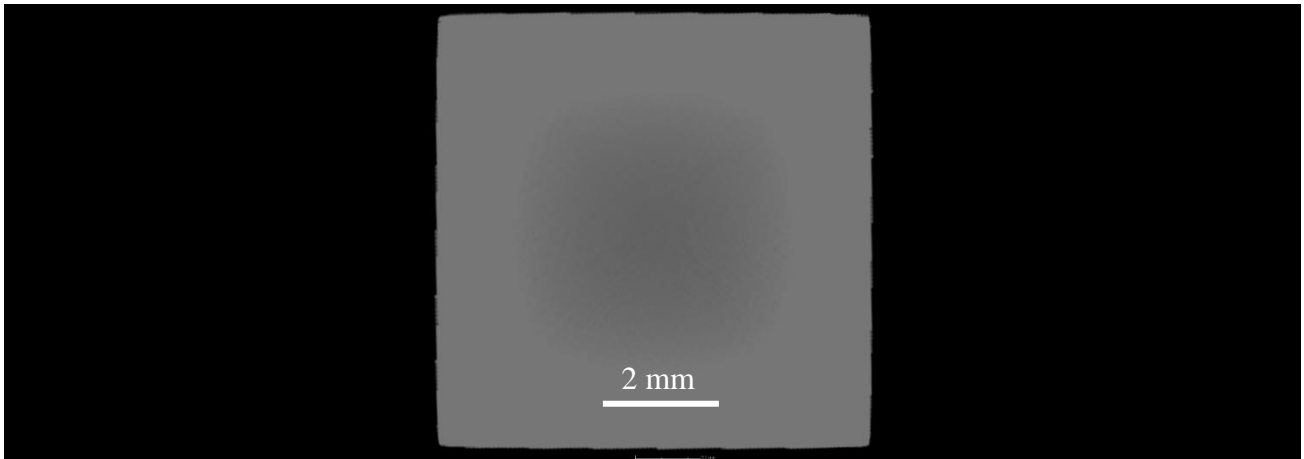
A reference artefact was designed to digitally simulate a CT scan of internal micro-holes, purposefully cut inside the material to simulate the presence of micro-porosity. Using a Computer-Aided-Design (CAD) model as an input to the aRTist simulation software (from Federal Institute for Materials Research and Testing (BAM)). Simulated XCT data was obtained to perform supervised training. Subsequently. Verification of the performance of this approach is compared with results from a commercial software.

## 1. INTRODUCTION

X-Ray CT is often considered as the future of non-destructive evaluation (NDE). It has the full capability of a 3D-scanning system to not only observe surface defects but also to visualize and quantify internal defects such as cracks and inclusions in a system. A typical industrial micro-CT system usually scans additively manufactured components with typical voxel and spot size of around 40-70  $\mu\text{m}$ . While this is a suitable system for detecting large pores above 0.1 mm, detection of micro porosity poses a challenge as the spot size is very near the resolution of the CT scanner. Nano-CT system manufacturers claim to have detectability of up to 0.3  $\mu\text{m}$ , but the system is known to be significantly more expensive. In addition, most of the time it requires a destructive method to be applied to the specimens to fit the penetration limit. Besides the problems arising from the resolution of the system, XCT has many other underlying sources of errors. (Texas

Geosciences, 2016) During scanning, lower energy photons get attenuated as they penetrate through a material, similar to a high-pass filter that resulted in an image with a darker center than the outer surface. Such a phenomenon is referred to as beam hardening. The shape of the sample itself too can lead to scattering artefacts, causing the image to get significantly noisier. The image (Figure 1) is an X-ray CT projection obtained from simulation with beam hardening artefact causing a non-homogeneous grey value display in the image: the central part of the image is darker than the outer part.

Due to these fundamental problems with X-ray imaging (Al-Shakhray & Al-Obaidi, 2003), a more adaptive method for defect detection is needed. This detection method should be able to detect small pores in the presence of scattering or beam hardening artefacts within the image. It is also important that the pixel-wise classification has to be as accurate as possible to the actual sample condition with minimum false detection or over-fitting.



*Figure 1. Example of X-Ray Beam Hardening. The central part of the image is darker than its outer part.*

The goal of this study is to explore a new methodology to accurately locate and quantify these defects to the nearest actual representation of the part's condition. Here the use of image segmentation method through random forest classifier was explored and an accuracy comparison was performed.

## **2. DATA SEGMENTATION**

It is easy for human brains to classify and segment an image as it works in a complicated way. Most traditional methods are based on simple intensity thresholding, or constrained models found by optimization (Arganda-Carreras, et al., 2017). This means that only a single image feature was used as the means to classify if an image falls under pore or material category.

Trainable segmentation enables a computer to extract multiple image features (e.g. edge information, pixel intensity) and classify the images based on a trained model that is set on past data.

The model is built on a classifier, and there are many image processing classifiers that are available to use. In this study, a random forest classifier method was used as the means to classify every pixel in the images into three classes; pore, air, or material.

### **2.1 Classifier**

Random forest is a well-established method for data classification and has been around for a long time (Ho, 1995). This classifier has been used in a wide range of industry and image processing field as a well-known image segmentation technique.

The random forest classifier works by generating multiple decision outcomes by bootstrap aggregating, or more well known as “Bagging”. This is a mathematical method that takes the original data and creates different sets of decision trees at random. This method allows the same data to be placed in a single bag in repeat. These “bags” of decision trees then vote for the likely outcome of each instance. The votes determine the overall classification identity (Breiman, Random Forest, 2001). The working concept of this classifier is illustrated in a diagram shown in Figure 2.

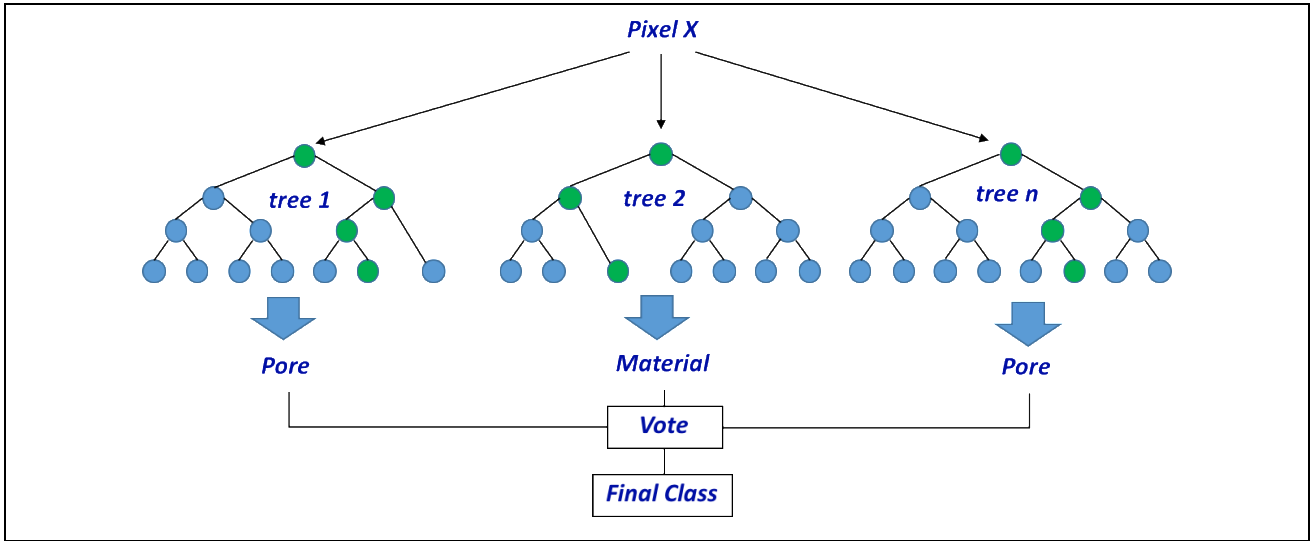


Figure 2. Random forest concept

## 2.2 Feature Extraction

The random forest classifier requires instances to be pre-defined. These instances were extracted by image processing through ImageJ/Weka filtering toolbox (Kaynig, Schindelin, & Arganda-Carreras, 2018). The features, as explained in the Weka documentation that was extracted are as follows:

### 2.2.1 Gaussian Blur

Gaussian blur filter extracts a blurred image using  $n$  individual convolutions with Gaussian kernels with the normal  $n$  variations of  $\sigma = 1, 2, 4$  and  $8$ . This operation is performed to blur the image until the pixels are homogeneous.

### 2.2.2 Hessian Filter

Hessian filter calculates a Hessian matrix  $H$  of a Gaussian blurred image. Hessian filter extracts the change in intensity of each pixel within the image in the horizontal, vertical, and diagonal directions, and compiles it into a  $2 \times 2$  matrix as shown in equation 1.

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (1)$$

By default, Weka extracts the modulus, trace, determinant, first and second eigenvalue, orientation maximizing the second derivative, gamma-normalized square eigenvalue and the square of gamma-normalized eigenvalue difference of the Hessian matrix. The details of these features can be found in Weka-ImageJ documentation (Arganda-Carreras, Trainable Segmentation, 2018).

### 2.2.3 Membrane Projections

In order to distinguish similarly-valued pixels that are different from the average intensity, a membrane projection filter was used to amplify membrane features within the image through directional filtering. It is defined as a 19x19 zero-filled matrix with the middle column entries set to 1. Multiple kernels are created by rotating the original kernel by 6 degrees up to a total revolution of 180 degrees generating 30 kernels. Each kernel is convolved with the image and then the set of 30 images are Z-projected into a single image via the sum, mean, standard deviation, median, maxima, and minima. Each of these numbers is placed as features in the overall random forest structure.

### 2.2.4 Sobel

This filter converts the gradient of the image to extract the edge feature based on a Gaussian blurred image. The Sobel operators are as follows;

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

### 2.2.5 Difference of Gaussian

This feature is useful to quantify the amount of blurring due to the Gaussian operator to determine the noise within a feature, it works by calculating two Gaussian blur images from the original image and subtracts one from the other.

## 3. REFERENCE ARTEFACT AND IMAGE ACQUISITION

### 3.1 Reference Artefact

A reference artefact was designed using Siemens NX CAD software in three different shapes and five different porosity volume. Hence a total of 12 designs were drawn for this experiment. The design of the porosity location was made in a circular location to simplify the measurement process. Five small cylinders were cut-extruded symmetrically inside the material to create voids with diameters ranging from 0.04 – 0.2 mm. The purpose of this variation is to quantify the difference in reliability of the measurement method when it is near and below the pixel size and also well above the pixel size.

The design of the reference artefacts is visualized in Figure 3. Image denoted as A, B, and C are the top-view cross section, of the center of the artefact. The inner circle represents the pitch-circle where the defects are located. I

The 3D drawing denoted in D, E, and F are the isometric, 3D representative of the artefact. The artefact D was designed in a parallelepiped shape, to simulate a high aspect ratio sample. Design E simulates a perfect XCT condition where all the penetration depth is the same in all rotation angles, and design F simulates similar aspect ratio circumstances.

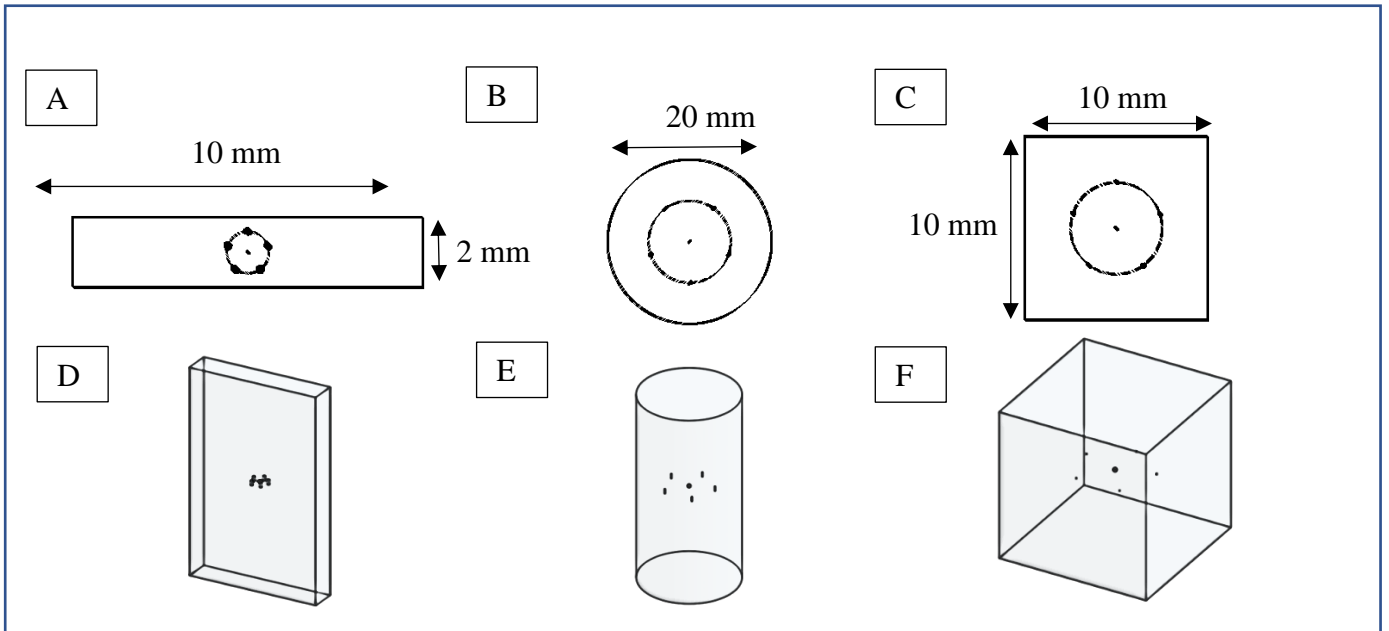


Figure 3. Reference artefact design

### 3.2 Image Acquisition

The software tool aRTist version 2 (Bellon, Jaenisch, Deresch, & Jaenisch, 2012), an X-ray Computed Tomography (XCT) simulation software developed by Federal Institute for Materials Research and Testing (BAM) was used to generate the simulated image. The reference of the settings is based on the Nikon XTH 225 ST X-ray machine specification with a Monte-Carlo scattering option enabled to a more accurate representation of the XCT image in real operating conditions.

The settings under which the simulation was run is as follows:

Table 1. aRTist XCT simulation parameters

Parameter Name	Parameter Value
Detector Resolution (mm)	0.2
Voxel size (mm)	0.067
Spot type	Point
Detector Size (mm)	410 x 410
Detector Type	Flat Panel
Source-Object-Distance(mm)	156
Sample Orientation (I,J,K)	76.2, -86.3, 78.4
Source-Detector-Distance (mm)	1200
X-Ray Beam Spectrum	General
Accelerating Voltage	200
System maximum voltage (kV)	225
Current(mA)	1000
Power (W)	225
Copper filter thickness (mm)	0.5
Number of CT projections	200

## 4. IMAGE PROCESSING & DATA TRAINING

ImageJ (Schindelin, Arganda-Carreras, & Frise, 2012) is a graphical user interface (GUI) open source image processing software with filtering toolbox with scripting ability that allows users to automate the filtering function and Weka is a machine learning software developed by the University of Waikato that is embedded into ImageJ image segmentation module.

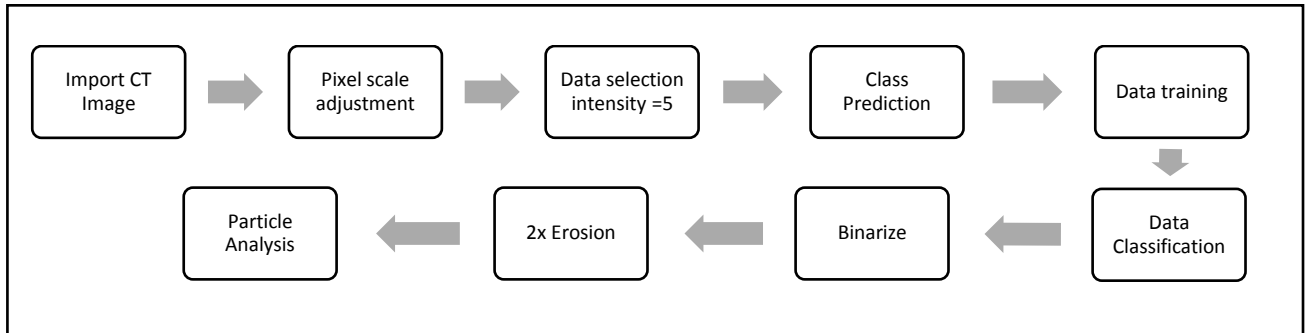


Figure 4. Image processing steps in Weka to analyze the particle size

Once the data is fully trained, the classifier was then applied to a new set of images. To do this, the data needs to be binarized. Hence, a dark output image with probability value lower than 0.5 was homogeneously assigned to 0, and those with probability equal and above 0.5 assigned to 255.

The binarized data then undergoes a pixel-erosion operation twice subsequently particle analysis is used to analyze the area that each island of lit pixels (here representing either edge or porosity) occupies. Figure 5 shows the ImageJ macro for this process. It is important to note that the edge is still visible in this experiment and was removed manually by selection tool to avoid misinterpretation of the particle analysis.

```

"/classifier.model");
call("trainableSegmentation.Weka_Segmentation.getProbability");
selectWindow("Probability maps");
run("Duplicate...", "duplicate channels=3");
setAutoThreshold("Default no-reset");
//run("Threshold...");
run("Convert to Mask", "method=Default background=Dark calculate black");
run("Erode", "stack");
run("Erode", "stack");//Remove Edge
run("Analyze Particles...", "size=0-1 pixel display exclude clear include stack");
  
```

Figure 5. ImageJ macro

In order to obtain the volumetric representation to the data, every detection pixel was summed with the entire stack of images and then multiplied by the thickness of the slice as represented in equation 3.

$$Vol = \sum_1^n Area * Voxel Size \quad (3)$$

The commercial software tool VolumeGraphics VGStudioMax was used as a benchmark for the accuracy detection as it is currently regarded as an industry-standard, off-the-shelf solution capable of performing many types of XCT image analysis such as pore detection. The software provides a pore detection toolbox with a proprietary VGDefX searching algorithm as its main, default defect searching algorithm. The parameters set for the analysis are tabulated in Table 2.

Table 2. VolumeGraphics VGStudio MAX settings

Parameter Name	Value
Algorithm	VGDefX
Probability	0.5
Min – Max volume	1 – 999999 pixels
Surface Determination	ISO-50 (advanced, remove void)

## 5. RESULTS

The pores detected using Weka trainable segmentation and Volume Graphics were compared to the original image with the exact predefined pore location (add figure reference here). The nominal diameter for the pore was designed to be 0.1 mm, with simulated an isotropic voxel size of 0.067 mm.

The images suggest that VolumeGraphics manages to spot the defect, however, the spotted area is much larger than the actual porosity size. The blue spots in the image are false detections which can easily be filtered. The detection area is much larger when compared to the original image. VolumeGraphics was not able to detect the smallest pores with a diameter equal to 0.6 voxel.

The Weka trainable segmentation method shows good defect approximation with minimal loss of accuracies in this case. There was also no false detection found within the image.

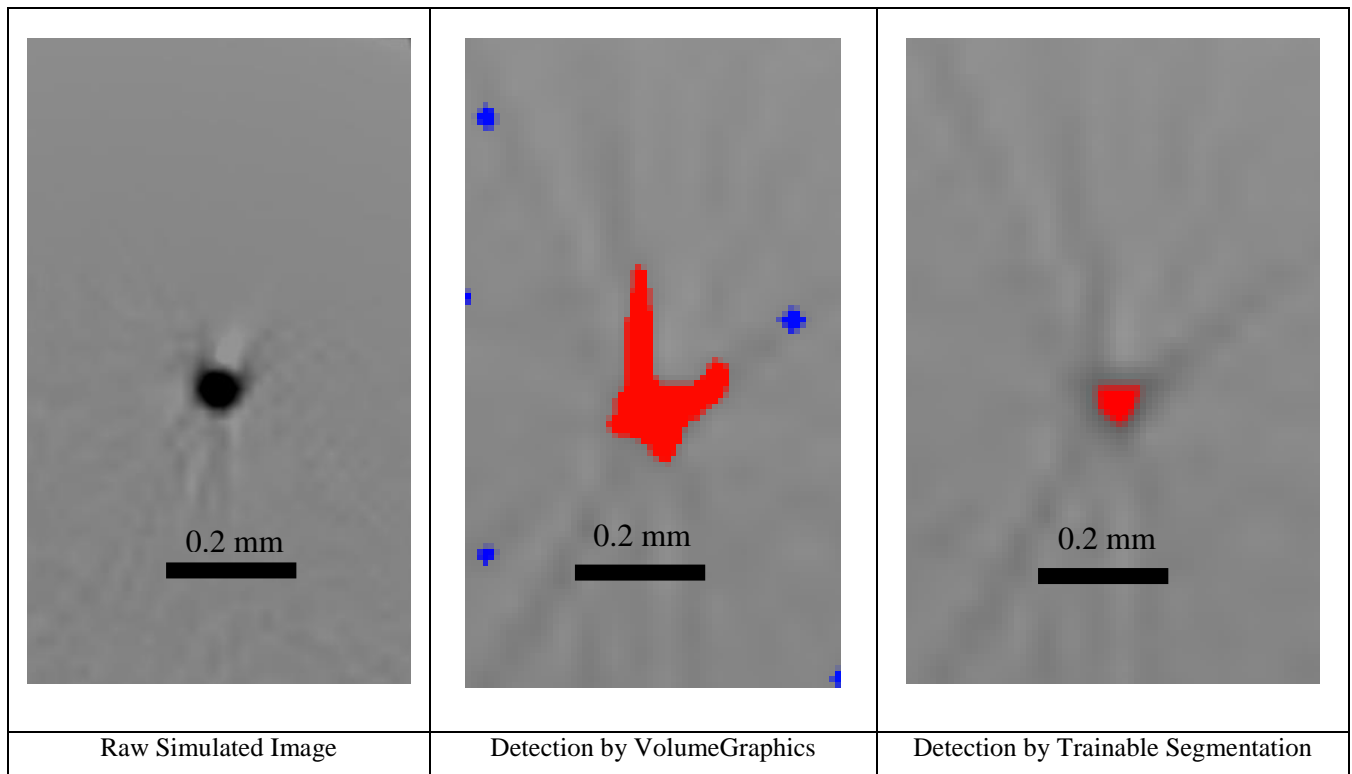


Figure 6. Visualization of pore detection using VolumeGraphics and Weka trainable segmentation



The accuracy study demonstrates the reliability of the detection method for different pore sizes. The detection test was performed on all test samples using five different pore sizes pre-set in the simulations (0.04, 0.07, 0.1, 0.15 and 0.2 mm). The simulation results were run on the VGDefX and trainable segmentation algorithms to establish the differences in detection accuracy.

The result of accuracy study was presented as a graph in Figure 7. The graph is plotted as percentage ( $E$ ) error against defect size. The percentage error ( $E$ ) is calculated as the ratio of the actual detection area (using Weka trainable segmentation tool or VolumeGraphics) against the nominal area as defined in equation 3.

$$E = \left| 1 - \frac{\text{Detected Volume}}{\text{Nominal Volume}} \right| \times 100\% \quad 3)$$

In Figure 7, the percentage error for Weka segmentation tool (in blue) and for VolumeGraphics (in orange) are decreasing with increased defect diameter.

Both segmentation methods have an important percentage error; for small defect diameters (0.6 – 1.1 voxel size) Weka segmentation tool has its  $E$  comprised between 41% and 830% and for VolumeGraphics the percentage error is much higher with  $170\% < E < 3650\%$ . The percentage error value varies also according to the shapes of the artificial reference samples.

For small defect diameters, Weka based method showed less variability between the three different shapes compared to VolumeGraphics's.

VolumeGraphics was unable to detect the defects in the cylindrical shaped sample; furthermore, the percentage error for flat and cube-shaped samples are spreading from 1300% to 3700%.

Increasing the defect size decreases the percentage error for both methods. When the defect is of three voxels size, Weka method reduces its error percentage to 100%. VolumeGraphics remain to have the highest errors compare to Weka's method with an error of 300%.

Traditional image segmentation techniques often require good contrast images, thus making difficult or impossible to accurately classify micro porosity near the X-Ray detector resolution.

For machine learning approach, such as Weka's method presented in this study, there is greater potential in detecting defect near or below resolution compare to VolumeGraphics.

Trainable segmentation approach gives much more accurate result in all point of the experimentation with minimum parameter adjustment. The effect of sample shape (cube, flat and cylindrical) on the percentage error is less noticeable in Weka for small defect compared to VolumeGraphics.

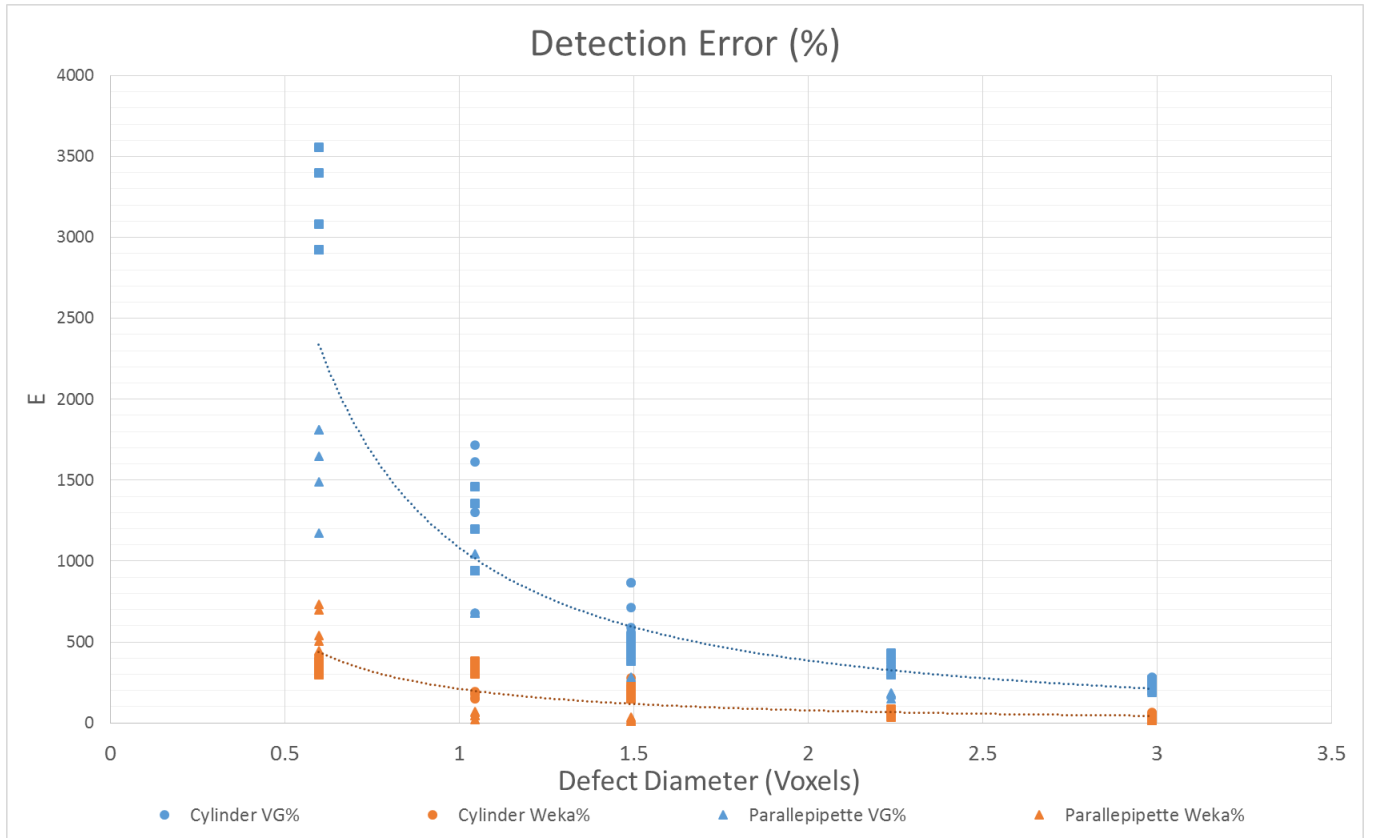


Figure 7. Detection error of trainable segmentation vs conventional software

## 6. SUMMARY & FURTHER RESEARCH

Trainable segmentation method showed promising results in detecting pore and defects near resolution in different sample configuration.

A Further test should be performed by varying the material of the sample or reducing the number of projections for example. The effect of sample shapes on detectability should further be studied, by increasing the number of samples and pores within a sample. To correlate the simulation result to the real-world scenario, it is also important to verify the result with a real XCT experiment on a real fabricated sample.

It is also known that more accurate approaches such as deep convolutional neural network could offer a better alternative to the machine learning approach presented in this study. Trainable segmentation methods such as the random forest approach have some limitations: the edge is accounted as a part of the defect and thus require manual intervention from the operator. Deep learning approach may be the solution to this issue.

## 7. REFERENCES

- Al-Shakhrah, I., & Al-Obaidi, T. (2003). Common artifacts in computerized tomography: A review. *Applied Radiology*, 25-30.
- Arganda-Carreras, I. (2018, June). *Trainable Segmentation*. Retrieved from ImageJ: [https://imagej.net/Trainable\\_Weka\\_Segmentation](https://imagej.net/Trainable_Weka_Segmentation)
- Arganda-Carreras, I., Kaynig, V., Rueden, C., W Eliceiri, K., Schindelin, J., Cardona, A., & Sebastian Seung, H. (2017). Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 2424–2426,.
- Bellon, C., Jaenisch, G.-R., Deresch, A., & Jaenisch, G.-R. (2012). Radiographic Simulator aRTist: Version 2. *WCNDT2008* (p. 7). Durban: NDT.net.
- Biau, G. (2012). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 1063-1095.
- Breiman, L. (2001, January). *Random Forest*. Retrieved from Berkeley: <https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf>
- Breiman, L., & Cutler, A. (n.d.). *Random Forest*. Retrieved from Berkeley: [www.stat.berkeley.edu/~breiman/RandomForest/cc\\_home.htm#mislabel](http://www.stat.berkeley.edu/~breiman/RandomForest/cc_home.htm#mislabel)
- Ho, T. (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, (pp. 278–282). Montreal, QC.
- Kaynig, V., Schindelin, J., & Arganda-Carreras, I. (2018, July). *ImageJ Trainable Segmentation*. Retrieved from ImageJ: [https://imagej.net/Trainable\\_Weka\\_Segmentation](https://imagej.net/Trainable_Weka_Segmentation)
- Koehrsen, W. (2017, Dec 28). *Random Forest Simple Explanation*. Retrieved from Medium: <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>
- Ng, G., Jarfors, A., Bi, G., & Zheng, H. (2009). Porosity formation and gas bubble retention in laser metal deposition. *Applied Physics A*, 16-20.
- Schindelin, J., Arganda-Carreras, I., & Frise, E. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 676-682.
- Texas Geosciences. (2016). *ARTIFACTS AND PARTIAL-VOLUME EFFECTS*. Retrieved from UTCT: <http://www.ctlab.geo.utexas.edu/about-ct/artifacts-and-partial-volume-effects/>