

Technische Universität München

Chair of Media Technology

Prof. Dr.-Ing. Eckehard Steinbach

Master Thesis

Automated evaluation of standard defect sizes
for casting porosity

Author:	Irem Öztürk
Matriculation Number:	03677343
Address:	Feldmochinger Str. 37 80993 München
Advisor (TUM):	Prof. Dr.-Ing. Eckehard Steinbach
Advisor (TUM):	M. Sc. Michael Adam
Advisor (MTU):	M. Sc. Naomi Piera Bellomo
Begin:	01.04.2021
End:	30.10.2021

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, October 29, 2021

Place, Date

Signature

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of the license, visit <http://creativecommons.org/licenses/by/3.0/de>

Or

Send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

München, October 29, 2021

Place, Date

Signature

Abstract

Shrinkage porosity is one of the defects that occurs in the manufacturing of components during solidification and insufficient feeding. This can significantly reduce the life duration of components by causing fatigue stress. As a result, it is necessary to detect and evaluate the casting porosity. Typically, porosity evaluation has to be performed manually by experts. However, manual assessment is not reproducible method and time inefficient. Moreover, the task of determining whether different pores in a 2D cross section should be treated as a single connected pore in terms of an initial failure size for fracture mechanical assessment is difficult. Therefore, the used criteria such as the threshold distance between pores to be unified as a pore group or pore size metrics are needed to be examined, whether they are plausible and valid. Since only little work in automation of measuring defect sizes has been achieved so far, there is the demand to develop an automated tool for microstructural analysis of casting porosity.

In this paper, an automated defect size assessment method based on image processing techniques is presented. This method allows for a quicker, more cost-effective, and most significantly, more consistent evaluation of porosity. Furthermore, the used criteria is validated through 3D renderings of sample materials. The results of this thesis provide the groundwork for follow-up projects to evaluate porosity effects on the fatigue life of components.

Contents

Contents	ii
1 Introduction	1
1.1 Motivation	1
1.2 Goal	2
1.3 Overview	3
2 Related Work	4
2.1 Casting porosity and its effect on the fatigue life	4
2.1.1 The area model	5
2.1.2 The Feret diameter	6
2.2 Metallographic evaluation of porosity in blades	7
2.3 Image processing and clustering algorithms	9
2.3.1 Mathematical image processing (Non machine learning approach)	9
2.3.2 Traditional machine learning approach	16
2.3.3 Neural networks	18
2.4 Pore size analysis evaluation with 3D Examination	19
3 Implementation	20
3.1 Full system overview	20
3.2 Image & data collection	22
3.3 Preprocessing	22
3.3.1 Image type estimation	22
3.3.2 Denoising	24
3.3.3 Thresholding	24
3.4 Contour detection	27
3.5 Defect size estimation	29
3.6 Validation	40
4 Evaluation	43
4.1 Evaluation metrics	43
4.2 Results	44
4.3 Discussion	56

<i>CONTENTS</i>	iii
5 Conclusion	62
List of Figures	64
List of Tables	69
6 Abbreviations	70
Bibliography	73

Chapter 1

Introduction

1.1 Motivation

A superalloy – or high-performance alloy – is an alloy able to withstand high temperatures that would deteriorate other metallic materials industrially used such as steel or aluminum alloys. The term was first used shortly after World War II to describe a group of alloys developed for use in turbosuperchargers and aircraft-turbine engines that required high-temperature performance [GLK97]. Superalloys exhibit excellent mechanical strength and creep resistance at high temperatures, good surface stability, corrosion and oxidation resistance [MS14]. Due to these properties nickel-based superalloys have been widely used in the manufacture of advanced aero-engine blades and gas turbine blades [WLY⁺21]. In the aerospace industry, turbine blades from nickel-based superalloys are typically manufactured by investment casting.

Porosity occurs in cast solidifying metals and alloys due to negative pressures generated during solidification contraction, and pressure developed by gases dissolved in the molten metal [GSTM92]. This thesis looks only at the porosity in Ni-based superalloys, which represents an applicable method in general due to the similarity of pores in different metals.

The influence of casting defects on the mechanical properties of these thermally and mechanically highly stressed components is indispensable [BR4]. The evaluation of porosity of turbine components made of nickel-based superalloys constitutes a key criteria in quality assessment [Ros13] (see figure 1.1). In many cases, porosity is a key microstructural feature which has a strong effect on fatigue life of a cast component [GYLL04]. Therefore, detection and evaluation of the defect size and morphology of casting pores are requisite.

Commonly porosity evaluation has to be performed manually by experts [SYS00, HBPL16]. The key tasks of such a manual check contain estimation of potential defects – which would be fracture mechanically effective by clustering pores into pore groups – and measuring the biggest defect size. The selection of the images that should be ignored in further analysis

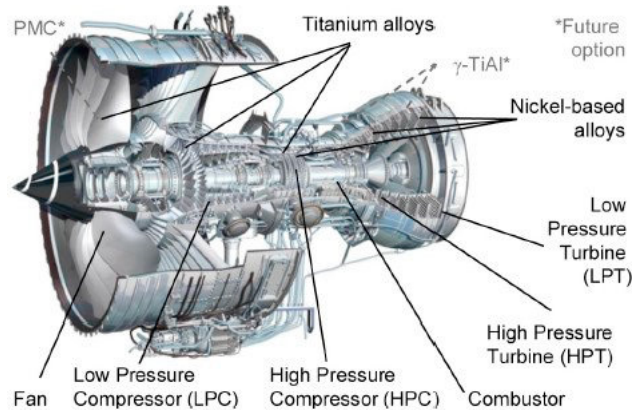


Figure 1.1: Distribution of titanium and nickel alloys in an aero-engine. The titanium finds its use in the cooler parts of the engine (fan and compressor), since it is stable to approximately 880°C . The nickel-based superalloys run in the hotter parts of the engine (combustor chamber and turbine), which can reach up to 1000°C [ICC⁺19].

according to their low porosity percentage is also such a key task.

However, this method has some drawbacks. It takes a huge amount of time and money to train and pay personnel for this task. Its performance vary from time to time and expert to expert and therefore it is not reproducible. Moreover, the used criteria such as the threshold distance between pores to be unified as pore group or pore size measures are needed to be investigated whether they are reasonable. The manual checks are done in two dimensional (2D) images and therefore the plausibility of defect size measurements acquired from 2D images needs to be investigated.

1.2 Goal

Since only little work in automation of measuring defect sizes has been achieved so far, there is the demand to develop an automated tool for the microstructural analysis of casting porosity (this will be outlined in Chapter 2). Defining the critical defect size is complex because of the inconsistency of 2D estimations and the different morphology of pores. Consequently, the manual assessments have uncertainties which need to be covered by further conservatism in the fatigue life predictions. This adds additional complexity to this problem when trying to automatically evaluate images based on ground truth images. Therefore, the thesis tries to answer the following research questions:

Research Question 1: Is it possible to develop a computer vision model to perform a robust and precise quantity analysis of pore space in Ni-based superalloys?

Research Question 2: Is the used criteria for estimating defect sizes of superalloy samples

plausible and valid when compared with slice-by-slice analysis on three dimensional (3D) segmented images of these samples?

1.3 Overview

The thesis is structured as follows: having introduced the topic in Chapter 1, Chapter 2 takes a deeper look into the current literature about pore structure analysis with its effects, metallographic evaluation of porosity and different image processing algorithms. After describing the main differences between state-of-the-art algorithms, Chapter 3 provides a detailed insight into the implementation of the automatic detection algorithm and the analysis of 3D results. Chapter 4 takes a critical look on how far the implementation was able to answer the research question by evaluating and discussing the resulting outputs. Lastly, Chapter 5 summarizes noteworthy observations inferred from the results and outlines future work in this field.

Chapter 2

Related Work

2.1 Casting porosity and its effect on the fatigue life

Porosity is defined as undesired irregularity such as any gap or hole in metal casting process. There are three main reasons why porosity might occur in casting superalloys [Vyn20]:

- a lack of feeding of melt to compensate for the solidification shrinkage;
- the rejection of gas, typically nitrogen, oxygen and hydrogen, from the newly formed solid phase to the surrounding melt;
- solid deformation as a result of tensile stresses, which can also be traced back to a lack of feeding of melt.

There are two main categories in the formation of porosity: gas porosity and shrinkage porosity, since the last reason can also be assigned as shrinkage porosity. The formation of gas porosity is based on the fact that the saturation for the important gases [H], [N] and [O] (formation of CO) in liquid steel decreases as the temperature drops, which is depicted in Figure 2.1 (see [RMB16]). On the other hand, insufficient development of the interdendritic flow lowers the pressure to certain point resulting in inadequate feeding and the creation of shrinkage porosity (see Figure 2.1). Since the shrinkage pores are more tortuous and larger than gas pores, their impact on the fatigue behavior is greater than gas porosity [LEH02, PGC⁺19].

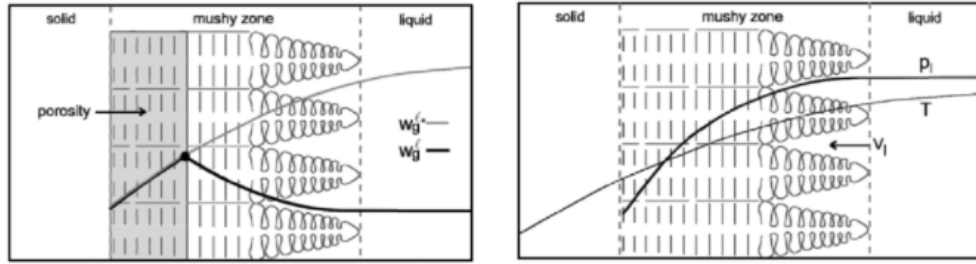


Figure 2.1: The image on the left-hand side illustrates the raising gas concentration w_g^l in the mushy region based on gas microsegregation and the diminution of the solubility limit w_g^{l*} . The image on the right-hand side depicts the liquid pressure drop in the mushy region based on solidification shrinkage. The parameter ρ_l is the specific mass of the liquid, T is the temperature and v_l is the velocity of the grain. The figures are taken from [RMB16].

The control of porosity in cast nickel-based superalloys is essential to extend the period of use – also known as service life – of jet engine components being operated at high-temperature under cyclic loads ¹ [PGC⁺19]. Fatigue of metals is characterized by the initiation and growth of cracks which grow in a self-similar fashion [WVD89].

The exact influence of a pore on fatigue behavior depends on several factors. For example, the pore shape and the distance of the pores to free surfaces and to other pores can influence the fatigue behavior. Furthermore, the size of the pores is an important factor. Defect sizes ranging from a few microns to several hundreds of microns can affect fatigue properties. It is complex to assess approximately the influence of several individual pores in a specimen. Nevertheless, the largest defect is often considered as the key parameter that governs the fatigue resistance [Mur85]. For the effective pore size of the materials in order to analyze their fatigue limit, some concepts are presented below.

2.1.1 The area model

Experimental evidence supports the concept that the critical stress for crack initiation does not constitute the fatigue limit of most materials [MB99]. Rather, it is affected by the threshold stress for nonpropagation of the crack which emanated from original cracks, defects or inhomogeneities. Therefore the stress intensity factor controls the critical condition for fatigue which is described by the following equation:

$$\Delta K = Y \cdot \Delta \sigma \cdot \sqrt{\pi \sqrt{area}}, \quad (2.1)$$

where ΔK is the stress intensity factor range, Y is the shape factor (0.5 for internal pore defect and 0.65 for surface pore defect as depicted in Figure 2.2) and $\Delta \sigma$ is the applied

¹Cyclic loading is defined as the continuous and repeated application of a load (fluctuating stresses, strains, forces, tensions, etc.) on a material or on a structural component that causes degradation of the material and ultimately leads to fatigue [Cor20].

normal stress amplitude (see in Figure 2.3). The threshold stress is correlated with the square root of the defect area i.e. \sqrt{area} projected onto a plane perpendicular to applied stress. This \sqrt{area} model is presented as size of the defect and easy to apply when compared with other models [ME94].

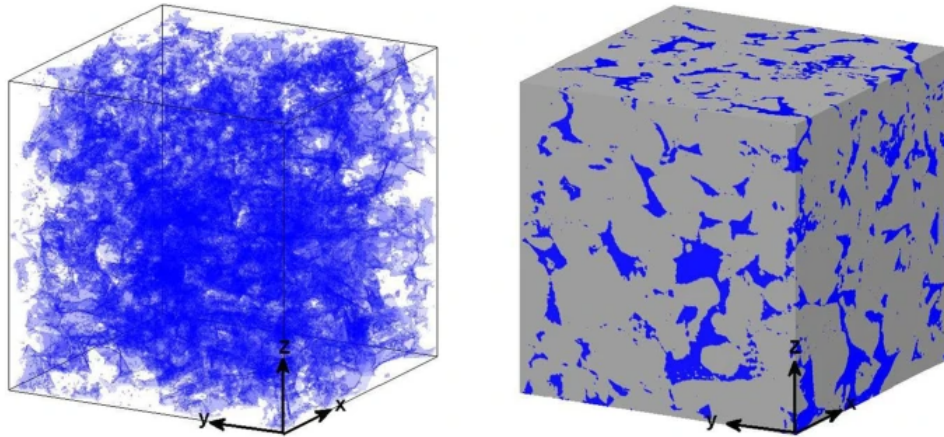


Figure 2.2: The image on the left-hand side demonstrates the internal pores, whereas the image on the right-hand side shows the surface pores. The blue marks represent porosity. The figure is taken from [WYZ16].

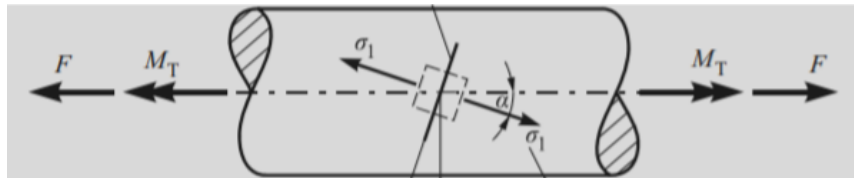


Figure 2.3: The component shown is simultaneously loaded by a force F and a torsional moment M_T . The fatigue crack grows perpendicular to the principal normal stress σ_1 [RS09].

2.1.2 The Feret diameter

The Feret diameter Φ_{surr} – the circumscribed circle diameter of a pore defect – is commonly used to describe the size of pore defects with complex shapes [NKF12, OSC⁺20]. Sphericity describes the shape of the pore defects and ranges from 1 to 0, showing that the pore defect becomes more and more irregular [WLY⁺21]. While the pore defect size increases, the shape becomes more irregular. The group with smaller pore sizes is named SP (Small Porosity), and the group with larger pore sizes is named LP (Large Porosity). Since the shape of large pores in SP samples is generally simple and the shape of large pores in LP samples is more complex, the Feret diameter is used to describe the size of the porosity defects [WLY⁺21]. A visual example of the Feret diameter is given in Figure 2.4.

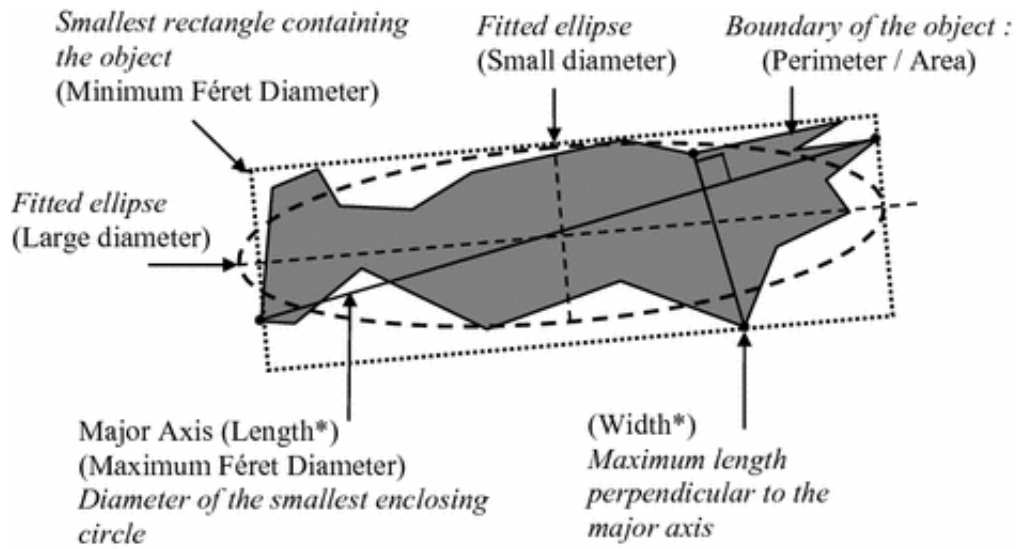


Figure 2.4: Lengths and widths of a pore are analyzed. Maximum Feret is also to be found on the figure [Pic17].

2.2 Metallographic evaluation of porosity in blades

As mentioned above, material inhomogeneities such as pores can reduce the service life and strength of a component under cyclic loading. Such that, a method is needed that can predict the lower limit of the service life. If the distance between two pores is less than a critical distance, it is expected that the crack nuclei corresponding to the pores will unite after a few load changes [ME94]. These pores are not considered as a single pore but as a pore network or a pore nest. A pore network defines several pores whose distances to each other are smaller than a critical distance. For the critical distances there are different assumptions such as:

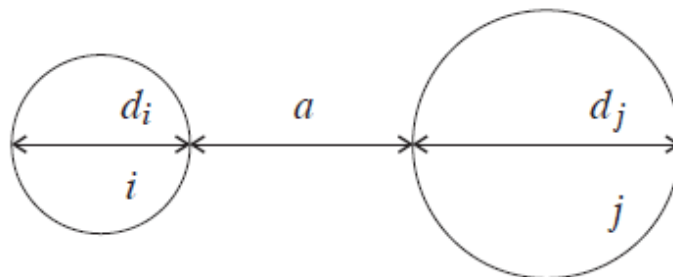


Figure 2.5: This figure shows two pores represented by the two circles i, j and the distance a between them.

- $a \leq a_{crit} = d_i$ [MB99]

- $a \leq a_{crit} = d_i + d_j$ [Kra13]
- $a \leq a_{crit} = 2d_j$

The circle i is the smaller pore and j is the bigger pore, d_i and d_j are the diameters of the pores, a is the distance between the two pores, where a_{crit} is the critical/threshold distance. Following on from this thesis, fatigue tests will be performed to determine the service life of specimens. For estimating the real lifetime of the components, the crack nucleation size of this pore or pore group needs to be calculated. In the following some suggestions in related papers are presented for defining the size of the pore network in 2D images.

Bounding rectangle

Finding the bounding rectangle around the pores which have the minimum threshold distance according to the given criteria is a recommended approach by MTU metallography experts and also used in [Kra13]. As the name implies, this approach finds the smallest rectangle which can contain all the pores in a pore network and the longest edge is taken as the defect size.

Pore areas

Murakami et al. proposed that the formula for calculating the crack nucleation size of a single pore from Section 2.1.1 can be applied to pore groups and be the size of the pore network [MB99, ME94, Mur85]. This results in calculating the sum of the pore areas in a pore network and taking the square of the result. The formula for the single pore size 2.1 is transformed into:

$$d = \sqrt{\sum_{i=1}^n A_i} \quad (2.2)$$

where d is the defect size or crack size, i is the index of a single pore, n is the amount of pores in the pore network and A is the area of the i -th pore.

Pore areas with imaginary pores

Another alternative approach for pore network size is the half of the sum of the pore areas and as well as the area of an imaginary pore [Kra13]. Since the distance between pores are not taken into account in the above approach, this method is proposed. First the pores of the pore nest are defined then the areas are calculated as above. Consequently, it is “imagined” that there is another pore between two actual pores which has the same diameter as the distance between them (see Figure 2.6). Then, its area is added to the

“real” pore area sum. Finally it is divided by two in order to reduce the extra amount of the imaginary pore. This approach has some drawbacks, which are presented in the Evaluation chapter.

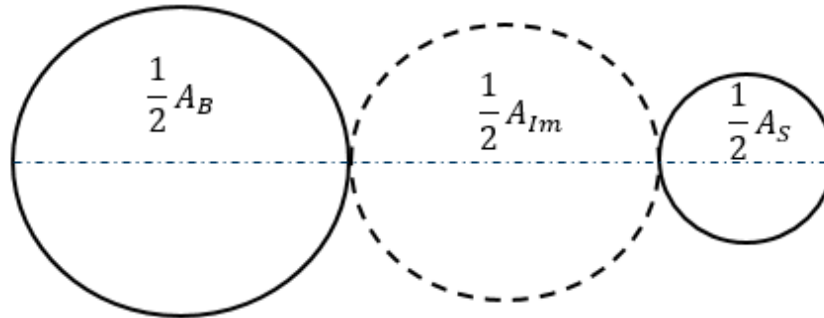


Figure 2.6: The left and right circles represent pores: A_B is the area of the bigger pore and A_S of a smaller pore. The dashed circle in the middle is the “imagined” pore which has the area A_{Im} . The method considers the half of each area.

2.3 Image processing and clustering algorithms

There are various approaches in image processing for performing microstructural analysis [CMY⁺17, RFB15, MBH⁺18]. In the following, some methods and applications are presented.

2.3.1 Mathematical image processing (Non machine learning approach)

Zafari et al. present a method for segmentation of partially overlapping objects with a shape that may be approximated using an ellipse [ZES⁺15]. The method starts with seed point extraction which is followed by edge-to-seed point association. Afterwards, the contours are fitted to ellipses. The experiments carried out throughout the research have proven the method presented as highly accurate in the detection and segmentation of overlapping objects. Their method achieves an accuracy up to 94 %. Smith et al. used the same approach and implemented an image processing method for a segmentation task using mathematical algorithms [SSS⁺19].

The results of the research by Campbell et al. demonstrates the potential of using image processing techniques for microstructural analysis of titanium components [CMY⁺17]. The authors segmented microstructural images of a Ti6Al4V alloy into its constituent grains and produce measurements with an automated method. Their approach is based on the Watershed algorithm outlined by Lalitha et al. with a range of pre- and post-processing steps [LAMS16].

Another approach is presented by Perzyk et al. involving a developed vision system [PŚ13]. It uses an advanced image processing algorithm based on modified Laplacian of Gaussian edge detection method and advanced lighting system. Their method contributes to inspection of surface casting defects.

Non machine learning methods are considered when a labeled “ground truth” training dataset is not available. It requires preprocessing such as denoising and thresholding and also more manual testing and adapting of the code than machine learning models [Sen20]. Edge detection is also an important preprocessing step for tasks including segmentation, active contours and object recognition.

In the following section, some methods and algorithms which are considered in non machine learning approaches to perform pore detection and clustering pore networks are presented.

Denoising

Denoising is a fundamental task in image processing applications, since the majority of the images are degraded by noise. The goal of the task is suppressing the noise whilst keeping key image features (see Figure 2.7).

- *Local denoising*: Local mean filters take the mean value of a group of pixels surrounding a target pixel to smooth the image. For instance Gaussian blurring or median blurring perform local denoising. These methods use a Gaussian weighted average or median of the values to replace the pixel element in a small region around that pixel. Another approach is *bilateral filtering*. It is highly effective in noise removal while keeping edges sharp [Ope21a]. However, the operation is slower compared to other filters. Compared to Gaussian filters, bilateral filter checks if a pixel lies on the edge or not, and thus avoids blurring edges.
- *Non local denoising*: Non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms [CM05].

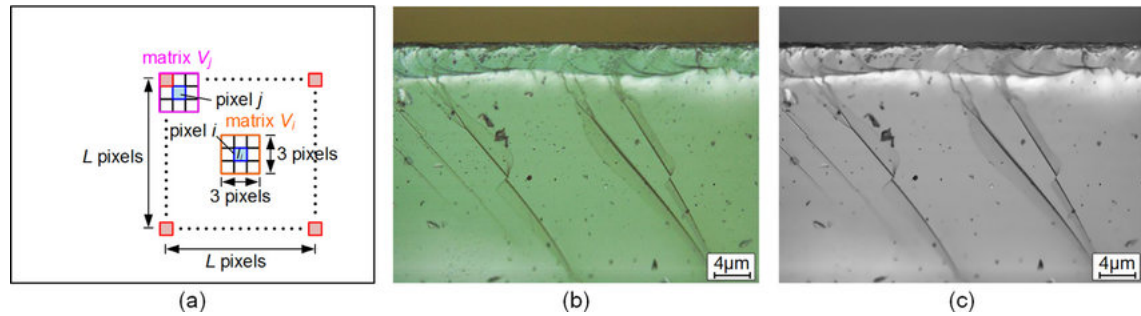


Figure 2.7: (a) Illustration of the *non-local mean denoising*, (b) the original and (c) denoised light micrographs containing subsurface cracks. The figure is taken from [ZYSL18].

As seen in Figure 2.7a, the basic principle of this method is as follows: for a certain pixel i in the image, each 3×3 pixel matrix (with the center pixel j , symbolised by “*matrix V_j* ” in Figure 2.7a) within the neighboring range $L \times L$ pixel region (denoted as the region Ω_i) is compared with the pixel i neighboring matrix (symbolized by “*matrix V_j* ” in Figure 2.7a) [ZYSL18].

Thresholding

In order to obtain binary images, the image processing operation of thresholding is required. Pixels are assigned 1 or 0 according to a set of criteria for gray scale values (see Figure 2.8).

- *Simple thresholding*: Also known as global thresholding, applies the same threshold value to every pixel. If the pixel value is smaller than the threshold, it is set to 0, otherwise it is set to a maximum value.
- *Adaptive thresholding*: The algorithm determines the threshold for a pixel based on a small area around the pixel. In this way, different thresholds are set for different regions of the same image, which gives better results for images with different lighting conditions in different areas [Ope20b].
- *Otsu thresholding*: This algorithm determines the threshold automatically from the image histogram. The mean value of the two gray scale peaks are calculated and set as the threshold value.

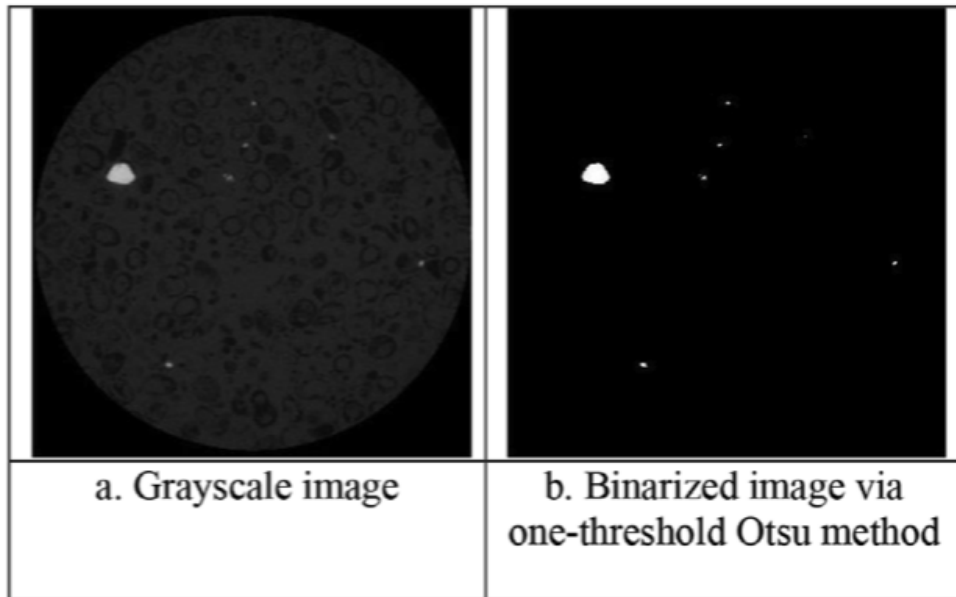


Figure 2.8: (a) Illustration of a gray scale image, (b) binary image obtained with Otsu thresholding algorithm. The figure is taken from [RIMB19].

Contour detection

In image detection applications, contour detection is a crucial step. The goal of contour detection is to identify the edges of objects on an image. An example of the steps that lead to contour detection is given in Figure 2.9. There are various methods for locating the borders of an object and the most common ones are listed below.



Figure 2.9: An example of an edge detection algorithm operation. The figure is taken from [Kho12].

- *findContours*: Suzuki et al. determined the surrounding relationship of the boundary of the image. The function *findContours* in opencv library depends on the algorithm outlined in [S⁺85].

This algorithm defines the hierarchical relationships between the contours and dis-

tinguishes between the outer and hole boundaries. Figure 2.10 depicts an example of an outer and inner boundary. The binary images are used as input. The algorithm iteratively checks pixels from the leftmost column to the rightmost and from the uppermost to the lowermost. When it evaluates an object pixel, it checks whether it belongs to the outer or inner border based on the following pixel type. OpenCV supports also only obtaining the outer contour of an object. The full code for this function can be found in [Ope20a].

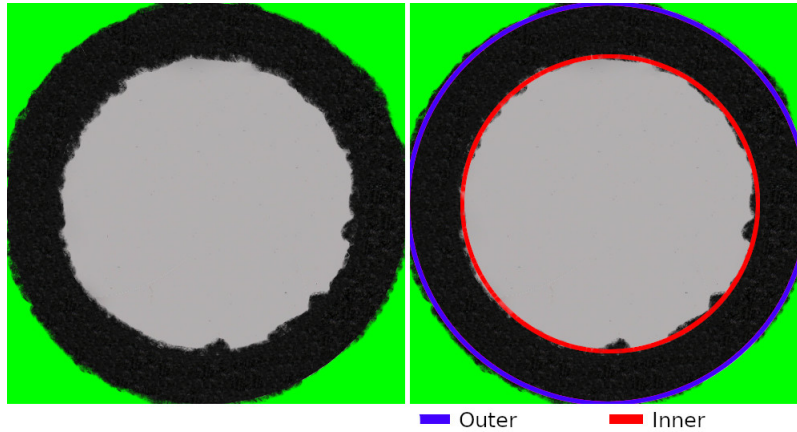


Figure 2.10: The left image shows the original image and on the right image the outer and inner contours are marked with blue and red colors, respectively. Figure is from [Tom20].

- *Sobel filter*: Sobel filter detects vertical edges (*Sobel X*) and horizontal edges (*Sobel Y*) independently and the sum of them (*Sobel X + Y*) contains all the edges. The gradient along x , and y directions are calculated using the convolution masks of 3×3 as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.3)$$

Here G_x and G_y are gradients along x and y directions. Gradient strength and direction of the edges are calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

$$\theta = \arctan(G_y/G_x) \quad (2.5)$$

- *Prewitt's Operator*: The difference between a Sobel filter and a Prewitt's operator is the used mask. Unlike the Sobel, this operator does not place any emphasis on the pixels that are closer to the center of the mask [Tsa18]. The mask is:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (2.6)$$

- *Laplacian filter*: Unlike the Sobel and Prewitt’s operators, the Laplacian filter uses only one kernel. It calculates two-dimensional gradients at once. Two commonly used small kernels are:

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

- *Canny edge detector*: In recent work, canny edge detection is one of the most popular image detection techniques [Uch13]. It is a nonlinear filter due to the nonlinear operations, whereas the above mentioned filters are linear. First, noise reduction is performed with a Gaussian filter. Using the Sobel or Prewitt’s operator, the gradient is computed. Afterwards, *non-maximum suppression* is performed, to remove pixels which are not considered as part of an edge. The final step is *hysteresis*, where upper and lower thresholds are used. If the gradient value of a pixel is higher than the upper threshold, then the pixel is considered as an edge pixel. Accordingly, if the gradient value is less than the lower threshold, then the pixel is rejected. If the gradient value is between the lower and upper thresholds, then the pixel will be accepted only if it is connected to a pixel that is above the upper threshold [RNR16].

Morphology correction

Binary images may have a variety of flaws. They contain noise and imperfections. Morphological transformations aim to overcome this issue by taking the shape and form of the image into account. An example image with a morphological correction is shown in Figure 2.11. They are performed with two main elements: image and structuring element i.e kernel. Some of the main morphology operations are:

- *Erosion*: A pixel in the original image (either 1 or 0) will be considered only 1 if all the pixels under the kernel is 1, otherwise it is eroded (made zero)[Ope21c]. Hence, the thickness or size of the foreground object decreases, the contour gets more thin.
- *Dilation*: Unlike erosion, dilation assigns 1 to the pixel if at least one of the pixels under the kernel is 1. It uniformly expands the size of objects [AAK00].
- *Opening*: Opening is combination of the two operations above “erosion followed by dilation” and used for removing noise such as small spurs and single-pixel noise spikes (high frequencies) by maintaining the original shapes and size of objects in the image [AAK00].
- *Closing*: The reverse of the opening operation, closing means “dilation followed by erosion”. As a consequence, small holes inside the foreground objects are closed or interruptions in the contour are corrected.
- *Morphological gradient*: Morphological gradient is the difference between dilation and erosion of an image. It gives the contour of a filled object as an output.

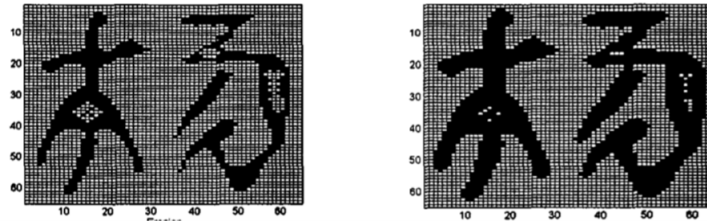


Figure 2.11: Implementation of dilation on the input image on the left-hand side. Figure is from [AAK00].

Defect size estimation

The main task of porosity analysis revolves around determining the defects in a material (see Figure 2.12). For the defect size estimation, it is needed to find the pores along a line or in one plane. Therefore one method is finding the bounding rectangle around the pores as mentioned in Section 2.2 and the other method is the *Hough transformation*.

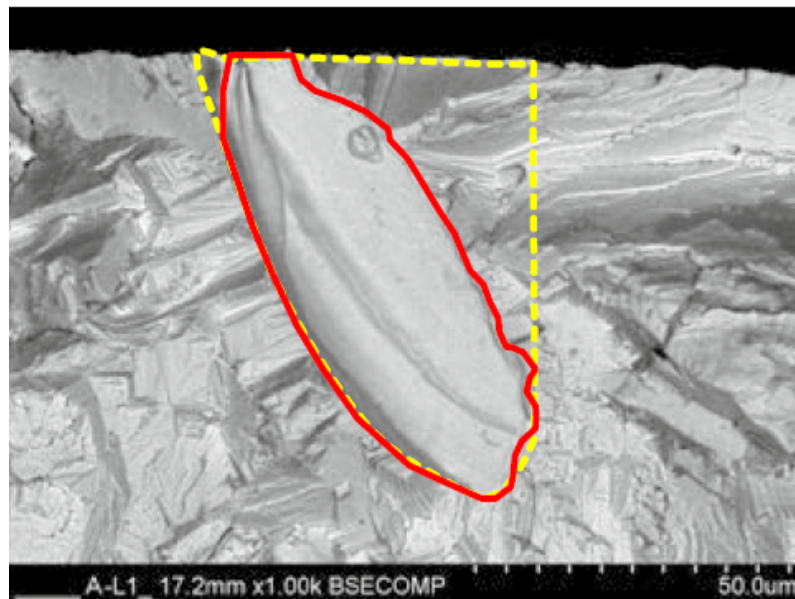


Figure 2.12: The red line marks the single pore in a specimen. The dotted yellow line shows the bounding right triangle. Figure is from [YMM⁺17].

- *Hough transformation*: Hough Transform aims to identify straight lines on a given image. One should perform first an edge detection before a Hough transformation in image processing. For a Hough transform, a line is expressed in its polar form. Hence, the equation of a line can be written as ([Ope21b]):

$$y = \frac{-\cos(\theta)}{\sin(\theta)}x + \frac{r}{\sin(\theta)}, \quad (2.7)$$

and with arranging the terms as:

$$r = x\cos\theta + y\sin\theta \quad (2.8)$$

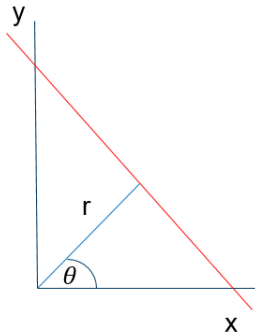


Figure 2.13: A line (red) can be represented in its polar form with another line which passes through the origin (blue). Its length from origin to the main line r and the angle of this line to the x-axis θ describes the line in polar form.

Each pair of points (x_0, y_0) can be substituted into the equation 2.8:

$$r_\theta = x_0\cos\theta + y_0\sin\theta, \quad (2.9)$$

where all possible lines that pass through this point are represented. The value ' r ' can be referred as resolution parameter *rho*. If one would plot all these lines, a *sinusoid* function appears. The Hough transformation performs plotting this *sinusoid* function for all the points in an image. If there are intersects, then those points are on the same line. In order to assign points as a line a threshold amount of intersects is set.

There are two types of Hough transformation: *standard and probabilistic*. Standard Hough line transformation performs the above steps and gives the result of vectors as an output. Probabilistic Hough transformation is more efficient and the output is the extremes of the detected line [Ope21b]. It also allows to change maximum line gap and minimum line length.

2.3.2 Traditional machine learning approach

Machine learning (ML) approaches generate solutions for classification and clustering problems in scientific images [DYL⁺17, WKH⁺18]. An overview to machine learning algorithms is given in Figure 2.14. Overall, these methods are examples of *Artificial Intelligence* (AI), which broadly “mimics” human behavior. Machine Learning is sub-branch of AI and *Deep Learning* (DL) is sub-branch of ML, which is explained in the following section.

Traditional machine learning models involve algorithms like *Support Vector Machine* (SVM) see [BP12], *Random Forest* (RF) see [Bre01], *Principal Component Analysis* (PCA) see [JC16]. The advantage of traditional machine learning models is that, the training set does not have to be as large as in neural networks.

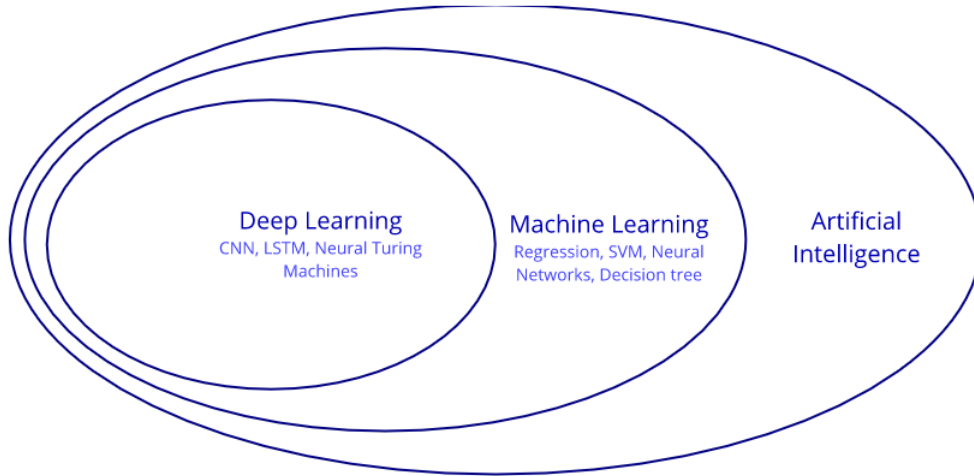


Figure 2.14: This figure highlights the relationship between Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL). Some example approaches of ML and DL are also given in the picture. Figure is from [O’R21].

Carter et al. attempted different approaches for the identification and segmentation of different microstructural elements and compared their performance [Sen20]. RF and SVM perform with 94 %, 91 % accuracy and a running time of 4s and 128s respectively in segmenting different microstructures.

K-means clustering is an example of unsupervised machine learning technique that identifies a predefined number of centroids and clusters ‘ a ’ data points close to a centroid by minimizing *inertia* (or squared Euclidean distances) to the closest centroid [Llo82]. The mathematical description of the stop criteria or inertia minimization can be described as below:

$$\sum_{i=0}^a \min_{\mu_j \in C_a} \|x_i - \mu_j\|^2 \quad (2.10)$$

where x_i are the data points within the cluster whose centroid is given by C_a and μ_j is the mean of all the data points in every cluster (see [SRRL⁺20]).

Singh et al. presented a novel application of contour detection, a computer vision-based technique coupled with k-means clustering, an unsupervised machine learning algorithm to distinguish between fractures and granular pores in a 3D segmented image [SRRL⁺20].

Their contribution is using also PCA to the contour features and then applying the k-means algorithm to the principal components (PCs).

The first output of PCA is PCs and the second is PC scores and PC loadings [LKA17]. PC loadings is a weight matrix that transforms the high-dimensional feature space into PC scores that can be projected onto the PC axes. These PC scores, z_k are given by [SRRL⁺20]:

$$z_{k(i)} = p_{(i)}w_{(i)} \quad (2.11)$$

where $k = 1 \dots m$ and $i = 1 \dots n$, w represents the weights that transform each row contour properties into PC scores and m stands for the total number of PCs. The used PCs are determined according to their information capacity, which is described by the *variance*.

Another method in estimating the orientation line of a pore network is Linear Regression [MMA20]. It tries to determine the best linear relationship between data points. It is usually used for survival prediction of components for estimating the life duration or in medical studies for the remaining life span of a patient [Pei20].

2.3.3 Neural networks

Deep Learning (DL) – or more commonly known as deep structured learning or hierarchical learning – is a division of machine learning which is based on a set of algorithms that attempt to model high-level abstractions in data [HYS16]. DL uses neural networks to analyze large data sets.

Many image processing methods benefit from deep learning algorithms. For example, Senanayake et al. (see [Sen20]) showed the best results in segmenting different microstructures with neural network in particular using a U-net structure with a modified, pretrained, residual, 50 layer deep, network (ResNet-50) [TAL16]. The running time for the prediction is one second, and the accuracy is 95 %. Hence, it is the best approach among mathematical and traditional machine learning methods. The U-net architecture achieves very good performance on very different biomedical segmentation applications as well [RFB15]. However, it requires thousands of labeled datasets.

Ma et al. proposed a novel deep learning-based image segmentation for microscopic images of Al–La alloys [MBH⁺18]. Using experimental results, the authors showed that this method outperforms existing segmentation methods such as K-means.

There is also a master thesis written at MTU, where a U-net structure is used to detect and segment microstructures in light microscopy images with nearly human-like precision [Eli18]. With this work, it is revealed that the U-Net is able to learn meaningful features without the need of manual feature extraction. The size of the training set is 2391 and the test set is 491 images. The executions are performed on a GPU to keep the training time at a reasonable level.

2.4 Pore size analysis evaluation with 3D Examination

Analyzing defect size on the 2D images is not adequate, since the observation is only restricted to one plane of the component. Therefore, 3D reproducible quantitative analysis of the materials are needed for the validation of 2D analysis tasks. Many of the recent techniques can not effectively collect large volumes while maintaining a high resolution, thus statistical evaluations are difficult to make [RNF18]. One solution for obtaining 3D microstructural data is serial sectioning.

Robo-Met offers an automated serial sectioning through a fully automated polishing station and a fully automated optical metallograph. It has been used to characterize the 3D morphologies of particles and pores in alloys [SMP03]. In its simplest form, the technique involves the repetitive grinding and polishing of the layers of the material. In each step, 2D images are taken which are then reassembled and rendered into 3D models. Spowart et al. achieved a slice thickness of $0.16\text{--}2.7 \mu\text{m} \pm 0.01 \mu\text{m}$ in a Ni-based superalloy [SMP03].

Lu et al. showed not a great agreement between 2D and 3D measures [LWW⁺20]. It is explained partly by the limited ability of 2D images dealing with features with a concave shape and partly to the limitations arising due to the projection of 3D information into a 2D plane.

Alternative approach for obtaining the 3D model is micro computer tomography (micro-ct). It offers the possibility to visualize pores in large volume of material and quantify their geometrical features and volume fraction using image analysis routines [HBPL16]. However, Robo-Met has two significant advantages for the type of work described here: (i) it relies for optical reflective contrast on etching rather than density and (ii) the specimen can generally be much larger [LWW⁺20].

Bright field microscopy aids in easy identification of pores due to large difference in reflection of light between metal and pores present in the sample and resolution is an order of magnitude higher compared to micro-ct which is useful in identification of small pores [GVG⁺17].

Chapter 3

Implementation

The algorithm was written in Python (version 3.7 see [Pyt18]) using standard *numpy*, *matplotlib*, *skimage*, *opencv*, *pil*, *math*, *scipy* libraries. The implementation was structured into five major steps: image & data collection, preprocessing, contour detection, defect size estimation and validation. In the following section, the full system overview is presented.

3.1 Full system overview

This section summarizes the workflow implemented for the analysis in Figure 3.1. The steps of the workflow can be assigned to three parts. In “Structural analysis”, the contour properties are detected and saved in a dataframe for further investigation. “Defect size estimation” has two main aspects: clustering and size estimation. Clustering means that the single pores are assigned to pore groups either according to their nearest distance (“Distance criteria”), or according to the grouping of the contour centroids (“K-means”), or aggregation regarding many contour properties such as perimeter, shape factors etc. (“PCA with K-means”). After defining the clusters, second step is estimating the size of the group either with “Bounding rectangle” algorithm or with “Pore area model”. The biggest group size is accepted as the defect size. There is also another approach “Hough transformation”, which is applied directly to the contours and does not require the clustering step. The size of the line determines the size of the defect.

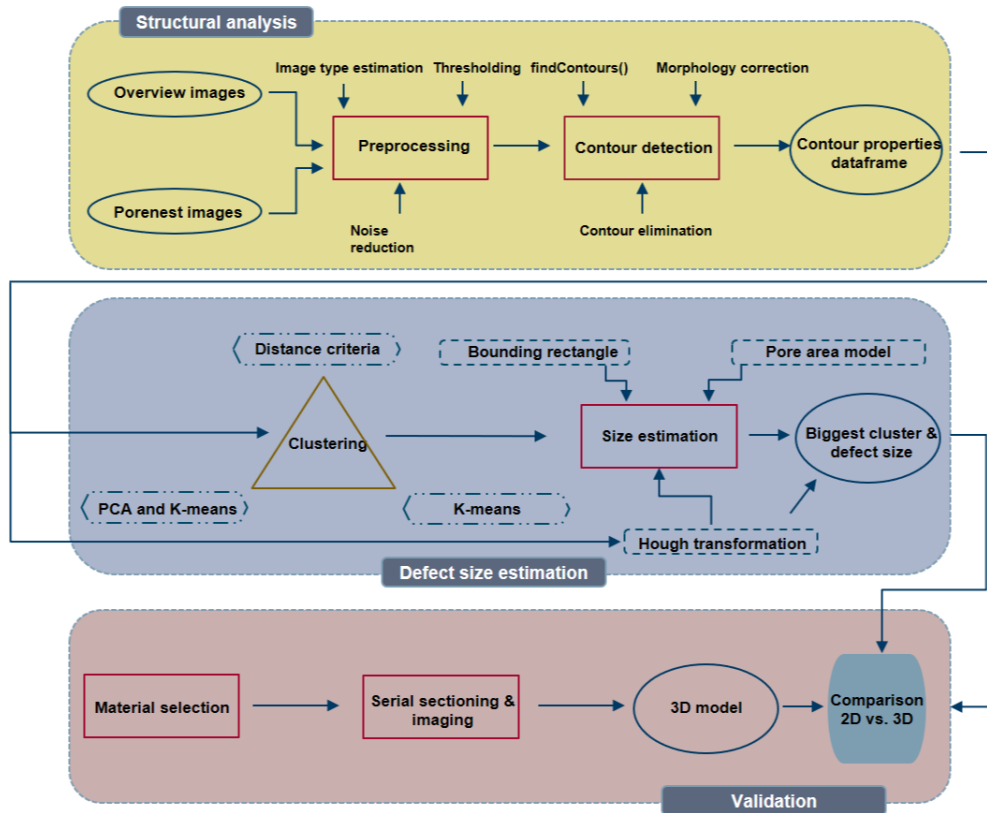


Figure 3.1: This figure shows overall pore analysis and defect size estimation workflow. The workflow is divided into three main parts: (1) Structural analysis of contours with the preprocessing and contour detection steps, (2) Defect size estimation with determining the biggest cluster and determining the defect size (3) Validation with obtaining the 3D model and comparing the results with 2D analysis.

3.2 Image & data collection

Firstly, images from the MTU internal database “*Image Management System*” (IMS) were extracted and used in this study. Under a light microscope, metal samples can only be viewed in incident light, since they are opaque [Für05]. Light microscopy allows magnifications by a factor of 1500x depending on the microscope [IS05]. In general, such magnification is sufficient to capture pores. They are clearly visible due to the high contrast from the surrounding area and appear dark and clear in light microscopic images. Also due to the increasing automation of microscopy, focusing, aperture adjustment and calibration are no longer a significant source of error. Therefore, this work considers only light microscopy images of turbine blades.

The MTU metallography department performs this process and uploads the images to the IMS. The images are labeled with a specific number assigned by the operator and the subfolders indicate the slice number that corresponding to different depths of the material. Metallography carries out the first analysis e.g. the percentage of pores. Then, these images are saved in the system and the porosity evaluator checks each image for a potentially critical defect. Those images might differ in calibration and also have different labels on them such as scale bars or some measurements on the image. For the automated analysis, images that should be considered are not allowed to have any text or scale bar. Therefore the images are saved without those additional features to a folder. For better tool development, the images which shows overviews of the total component are eliminated.

For defect size estimation, one should know the equivalent meter value per pixel on the images, which can be obtained from the metadata files (*XML* files) in IMS. To save the original image names and calibrations in a *CSV* file, a script is executed before the defect size estimating algorithm. In other words, the pre-step of running the tool is reading the meter per pixel value from *XML* file and saving with the original image name in a *CSV* file. During the execution of the program, this value is applied whenever a distance is measured in order to get the real defect size.

3.3 Preprocessing

A preprocessing step with the images is necessary to later use them in the program. This step contains an image type estimation, noise reduction and thresholding.

3.3.1 Image type estimation

The images in the collection for this study can be broadly divided into two main categories: overview images with a large field of view and detailed images of particular “pore nests” at higher magnification than the overview images (see Figure 3.2). Overview images shows

part of the turbine blade usually with a black background when imaged using a light microscope. “Pore nest” images result after the metallography checks the overview image and selects the region where the porosity density is rather high. They usually have a measure about 1 mm^2 .

Image type estimation is required, since different types necessitate different preprocessing procedures. For instance, the background in overview photos must be detected and ignored when measuring the porosity percentage. Furthermore, the threshold length of a contour for removing very small and very large pores differs. In order not to include the outer contour of the component in the analysis, the overview images have a smaller allowed maximum length for contours. The minimum threshold contour is also smaller, since they have smaller amount of pixels per pore. Also a closing operation is needed for overview images, in the cases where the pores conjugate the outer contour of the component. Then, contour estimation detects the outer contour, as well as the pore as one and the result is eliminated with the outer contour.

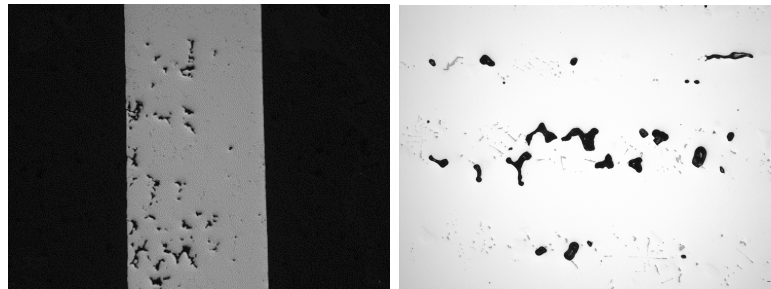


Figure 3.2: The image on the left is an example of an overview image, whereas the image on the right shows a pore nest image.

In order to differentiate between pore nest and overview images, one should take a look at the main difference which is the ratio of black pixels to white pixels. In general, overview images show higher ratio than pore nest images, since the background is also visible on the image. Based on that, the first approach is setting a threshold for the gray scale mean value of an image. To assign an image into a class, its grey scale value is compared to the set threshold value. 50 overview images and 40 pore nest images were analyzed. The result of the gray scale mean value of the overview images was 105 and of the pore nest images 217. The mean value of both taken as a threshold value which was 161. So when an image is read for the analysis, first the gray scale mean value is estimated from the whole pixels and if it is below 161 then it is assigned to the overview class, otherwise to the pore nest class. However, this algorithm results in a low accuracy (47%) with another dataset of 50 pore nest and 50 overview images. This occurs due to the different lightening conditions of images. Hence, various subsets have different mean values, and the range cannot be restricted to a single interval.

The second method is directly computing the ratio of the black values of all pixels. This method yields that 94% of the pore nest images have the percentage of black pixels below

10%. Therefore, the percentage method is implemented in the pipeline and the threshold value is set at 10%. Below that value the image is classified as a pore network.

3.3.2 Denoising

To reduce the background noise of the images, there are different approaches as outlined in the previous chapter. In order to determine the best suitable algorithm in the image set, both local and non-local algorithms are used. Non-local denoising approach yields better results in removing the noise. Two example images are shown in Figure 3.3, that were denoised with different methods.

In the scope of this study, a non-local denoising algorithm is applied on gray scale images. In order to perform non-local denoising, some parameters needed to be preset. Filter strength h regulates how strong the denoising is. Higher values of h cause some loss of details and smaller h preserve residual noise. In the implementation, visual tests show the best result with 30. Also the sizes of search and template windows needs to be estimated. The template patch is used to compute weights. Search window is utilized to compute weighted average for given pixel. Both window sizes are measured in pixels and should be odd. Greater size for search window causes greater denoising. An optimal setting for the images studied was 7 as template window size and 21 as search window size.

3.3.3 Thresholding

On various sample images, both global and adaptive thresholding were evaluated. The Figure 3.4 demonstrates that, global thresholding achieved better results in porosity clarity. On the other hand, adaptive thresholding causes noise in the image.

The challenge of global thresholding is calculating the threshold value at which a pixel's gray scale value should be allocated as black or white. Otsu thresholding is used to estimate that threshold for this purpose. In the image data collection, the gray scale values show two peaks in their image histogram, since foreground and background are distinguishable. Figure 3.5 shows the result for the sample images.

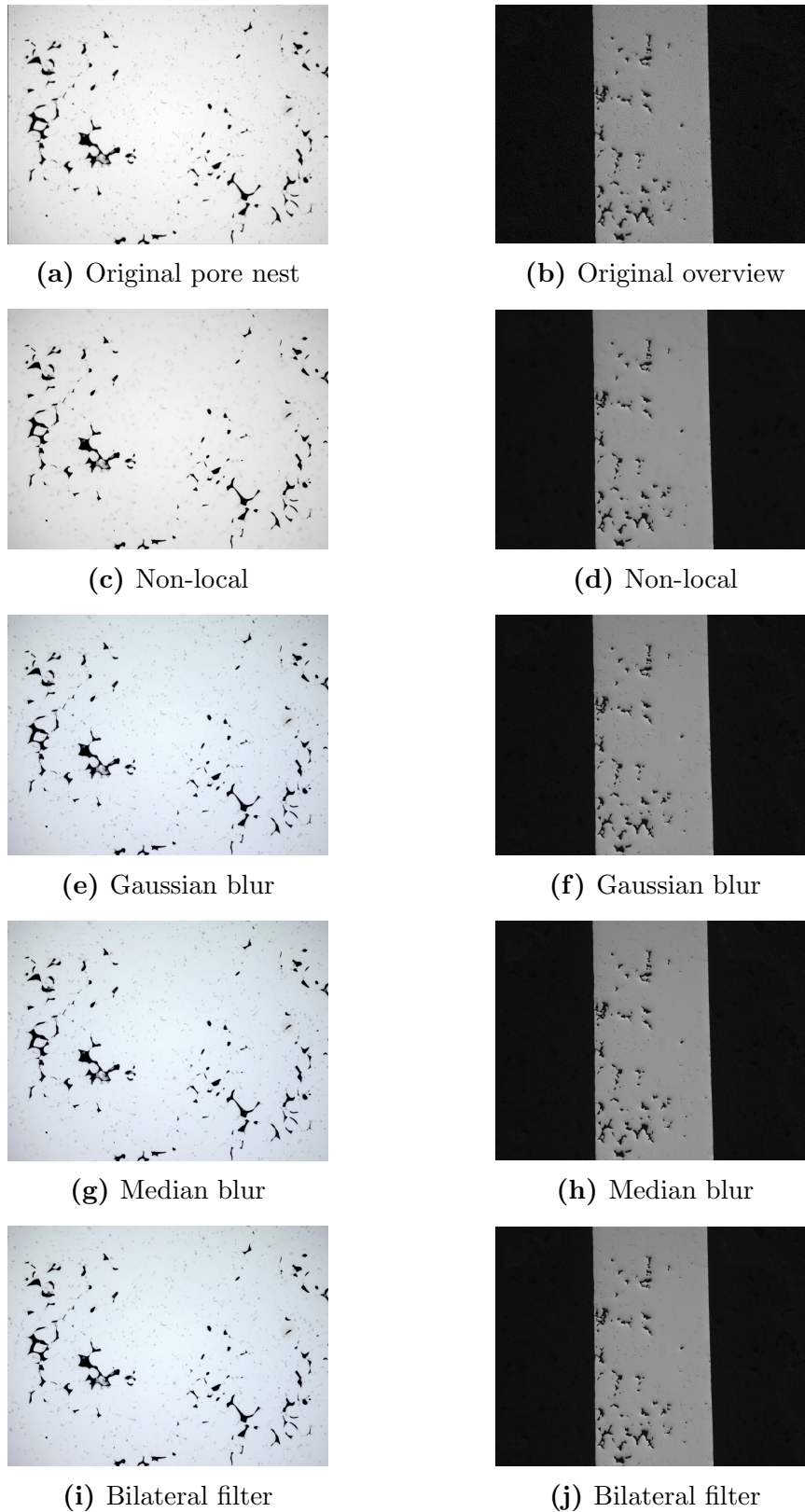


Figure 3.3: The images (c), (e), (g) and (i) show a pore nest image (a) denoised by different algorithms. The images (d), (f), (h), (j) show the same process on an overview image (b) accordingly. For the non-local denoising, h value is chosen as 30, template window size is 7 and search window size is 21. The used Gaussian kernel size here is (5,5), aperture linear size for median blur is 5. For bilateral filter is the sigma color 9, sigma space is 75 and border mode is 75. These input parameters are used in order to render the sample images with opencv library.

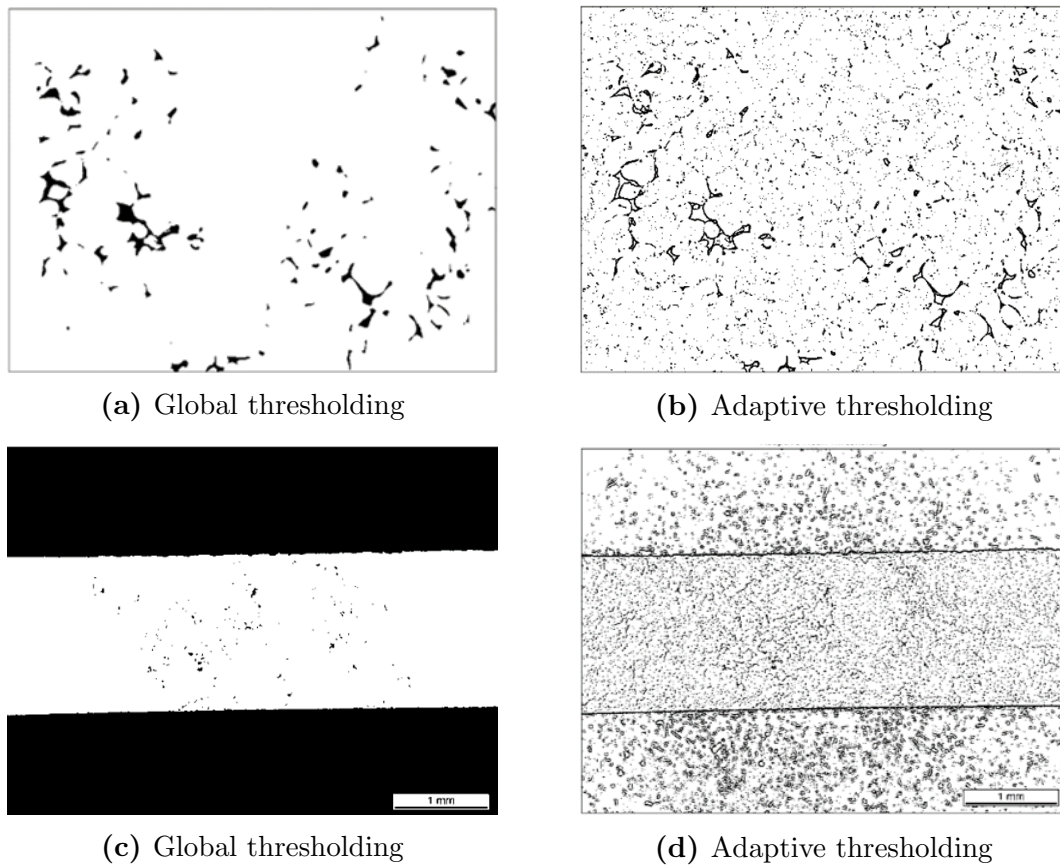


Figure 3.4: First row of images show an pore nest image both global and adaptive thresholding applied. Second row shows the same process on an overview image accordingly. For the global threshold 127 is chosen. For adaptive threshold, size of a pixel neighborhood that is used to calculate a threshold value for the pixel is 11.

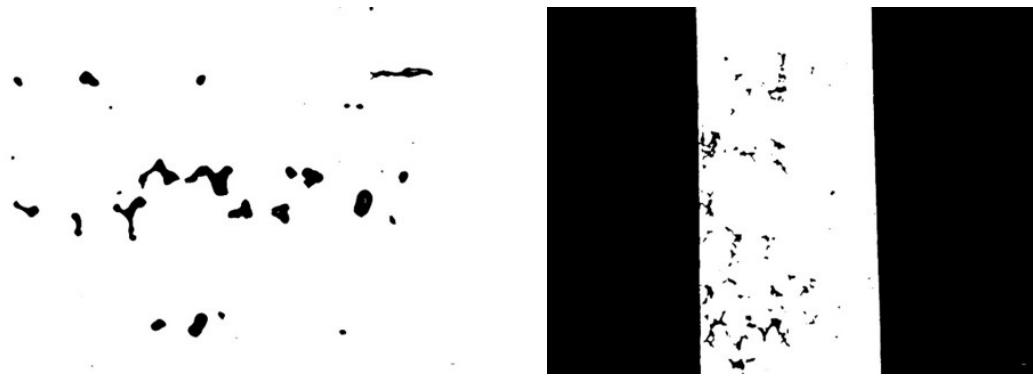


Figure 3.5: The image on the left shows a pore nest image with Otsu thresholding applied. Right image shows the same process on an overview image accordingly.

3.4 Contour detection

So far, the pipeline has prepared the images in order to have a better contour detection performance. This section revolves around constructing the contour map of pores. Figure 3.6 shows some example images with different contour detection algorithms.

Sobel filter, Canny edge detection, Laplacian filter and the function *findContours* from opencv library is applied to images. *findContours* has the best performance detecting the contours correctly and has the benefit of being able to use the implemented functions immediately without any other operation. Sample images are shown with detected contours in Figure 3.7.

It can be seen from Figure 3.7 that the outer contour of the metal is also detected and needs to be ignored in the following pore analysis. Therefore the maximum allowed area of a contour in overview images is adapted and set to 100 000 pixels. Then, the outcome appears to have eliminated some pores as well. This is caused by the fact that some contours conjugate with the outer contour of the component and elimination of outer contour results in elimination of those pore contours as well. Thus, morphology operations are needed. In general, morphology operations did not affect the contour detection performance according to visual tests. It is therefore applied only overview images, where pores may be detected together with the outer contour. The output image can be seen in Figure 3.8. In pore nest images, this does not cause a problem and the contours are mostly detected correctly since there is no outer contour of components to be seen on the images. According to their pixel ratio in pores and background the maximum allowed area of a contour is determined as 20 000 and minimum as 50. For overview images, the minimum value corresponds to pore nest images.

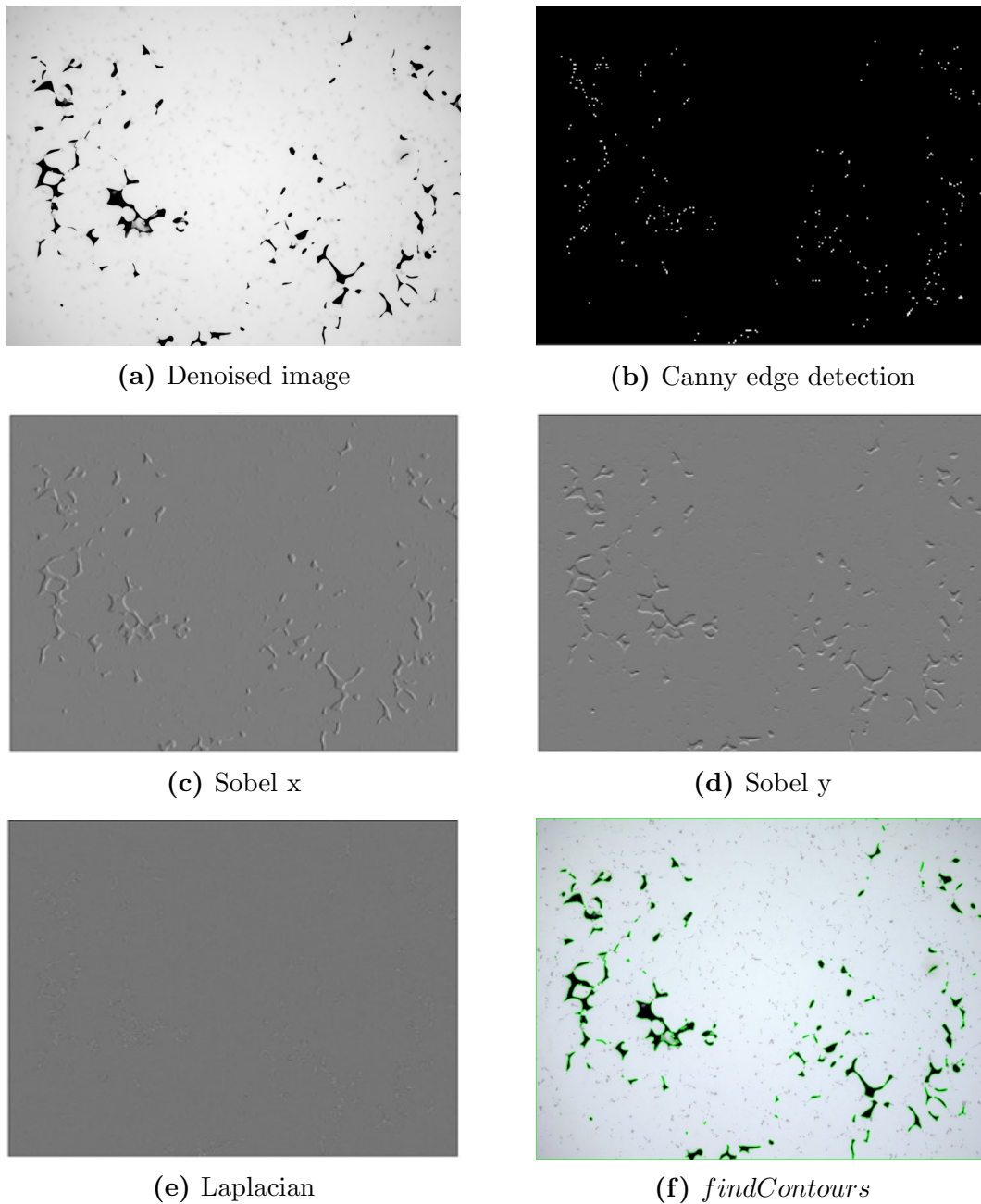


Figure 3.6: Different edge detection algorithms are applied to a sample image. For Canny edge detection, the first threshold for the hysteresis procedure is 100 and the second threshold for the hysteresis procedure is 200. For Sobel filters, kernel size is 5.

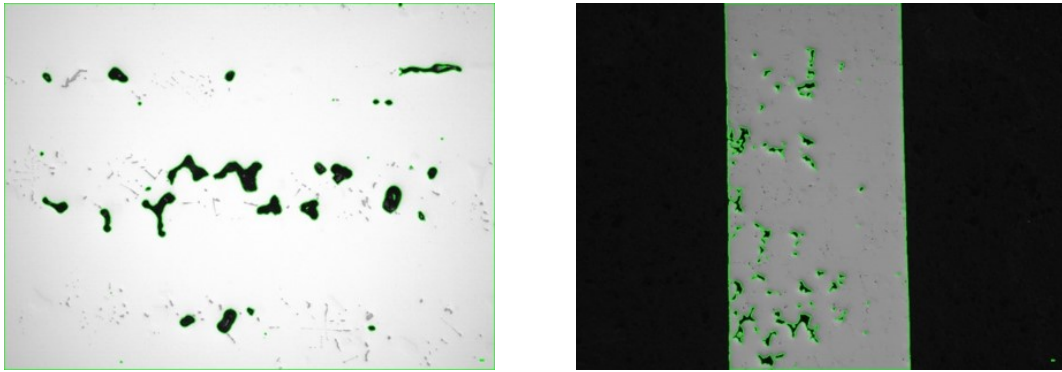


Figure 3.7: The image on the left shows a pore nest image with *findContours* function applied and the detected contours are drawn on the image. Right image shows the same process on an overview image accordingly.

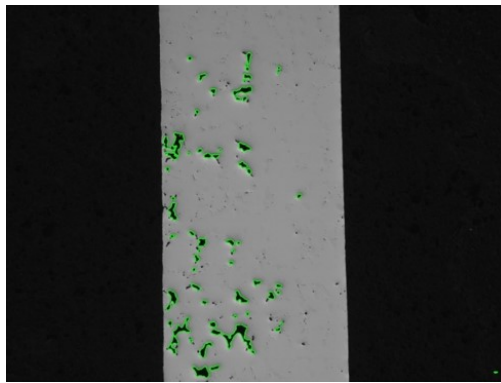


Figure 3.8: Closing morphology (dilation followed by erosion) applied to the image with the kernel size (8,8).

3.5 Defect size estimation

The approaches outlined in Section 2.3.1 are assessed before beginning to implement defect size estimation method, and some of them are eliminated. For instance, the defining pore networks with imaginary pores from Section 2.2 is not implemented, since it necessitates numerous calculations. In order to find the imaginary pore in the inter-pore area, the distances are taken into account. As shown in Figure 3.9, the pores in the perpendicular direction of the defect orientation might increase the measurements.

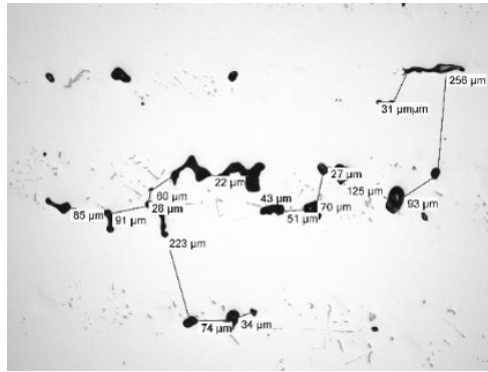


Figure 3.9: The pore nest image shows a pore nest with marked lines in inter-pore area. One can recognize the pores which are perpendicular to the main defect orientation.

Since the task of the thesis revolves around the maximum defect size and does not require segmentation between microstructures, supervised machine learning techniques such as SVM and RF models are not considered as useful for this use case. Moreover, neural network models are eliminated for the same reason. In addition, they require thousands of labeled data, which puts more time and effort in the analysis. Therefore, the intensive training of neural networks is saved.

Following the decision on the methodologies to be employed in the analysis, certain preliminary operations are carried out. Firstly, some major properties of contours are saved in a dataframe, so that they are not calculated repetitively in each step and can be read from that dataframe directly. These properties are the coordinates of the contour center, the contour area, perimeter and the radius of the circumscribed circle. Except for coordinates, other variables are all saved after they are calibrated with the previously saved calibration scale value, so that they correspond to the real measures.

Consequently, the distance matrix of the contours needs to be estimated in order to define pore network groups. After the contour detection was performed as described in Section 3.4, the distances between each contour pair are calculated. In order to increase the efficiency, the distance matrix is created as an upper triangular matrix, where the diagonal elements are also zero. So when a distance of one pair is saved, that pair is not checked again. To calculate the nearest distance between two contours, all pixels from these two contours are regarded and the smallest distance is saved.

For further analysis with mathematical models, the pore group needs to be estimated with the different threshold methods (see Section 2.2). Once the assigning of the pores into pore network groups is completed, the algorithms such as bounding rectangle and pore area model can be applied.

To measure the defect size of a pore group, different algorithms are applied. The implemented methods are: bounding rectangle (Section 2.2), pore area model (Section 2.2), Hough transformation (Section 2.3.1), K-means and PCA analysis (Section 2.3.2).

In the program, all methods are available for the user and can be implemented where needed for the defect size estimation.

Bounding rectangle

First model is bounding rectangle, which serves as an automated substitute to the manual approach, i.e. takes over the manual task. During manual check of an image, the expert tries to find the largest orientation line of pore network groups. To perform this automatically, the bounding rectangle method draws a rectangle around the assigned pore networks. The largest edge of the bounding rectangle is determined as the defect size.

To find the smallest bounding rectangle of pores, the first step is computing a convex hull (see Figure 3.10) of the points of interest. The edge orientations are calculated with the *arctan* function. Then, these orientations are used to rotate the convex hull and each time the minimum and maximum x and y values are computed. Consequently, these values are the axis of the surrounding rectangle. The rotation setting which results in the minimum area of the rectangle is stored as the operational output.

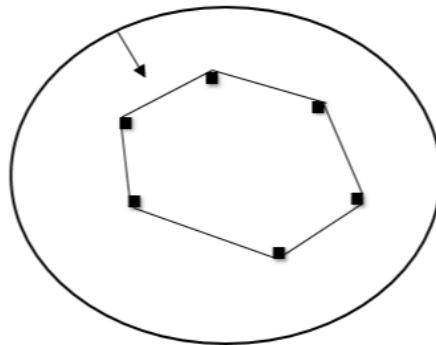


Figure 3.10: Convex hull of a bounded planar set.

Pore area model

For this model, a similar approach is used as previously outlined for the bounding rectangle model. First the pore network groups are assigned. Then, the biggest cluster is defined according to the largest sum of the pore areas in that group. Consequently, the defect size is calculated using equation 2.2 with the areas saved in the properties of the contours matrix.

Hough transformation

In this study, a standard Hough transformation does not provide reasonable results using the available image set, more specifically there is not a detected line in many cases. Hence, the probabilistic Hough transformation is applied in the pipeline.

The first visual results appear to be more akin to manual estimations with Canny edge detection than the edge detection algorithm included (see Figure 3.11 and for manual estimation result 4.10a). Therefore, the Canny edge detection algorithm is performed before running the Hough transformation for the images. For using the probabilistic Hough transformation function with opencv library, the resolution parameters ρ and θ are selected as 1 pixel and 1 degree, respectively.

In the conclusion of testing, the threshold parameter to assign a group of intersections as a line was found to be 50. The minimum allowed contour area 50 px^2 resulted in a minimum allowed diameter of 8 pixels. Correspondingly, the minimum line length is determined to be 10 pixels. The maximum gap between two points is also a free variable and can be set by the users when starting the Hough transformation in the program. This parameter is equivalent to the distance threshold in the bounding rectangle model. Reasonable results can be achieved with a minimum of 250 pixels.

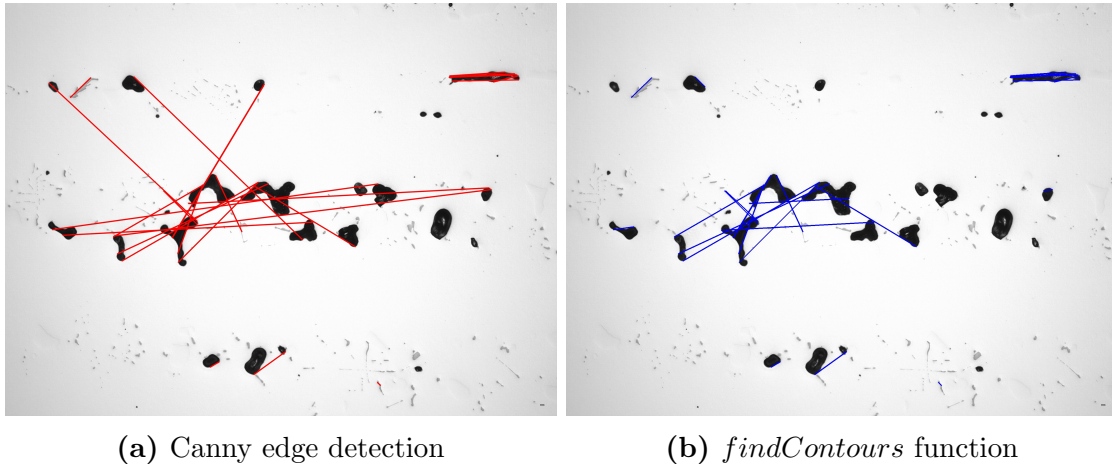


Figure 3.11: The image (a) is created with probabilistic Hough transformation after applying Canny edge detection algorithm. The image (b) displays detected lines on denoised images using the probabilistic Hough transformation and the *findContours* algorithm.

K-means algorithm

K-means clustering is an unsupervised machine learning technique that requires one predefined parameter, number of clusters ' r '. Changing the number of clusters can significantly enhance the performance of the K-means algorithm and therefore, this number needs to be

fitted. This fitting can be combined by the Silhouette score analysis. The Silhouette score assesses the distance between clusters and the likelihood of a data point being assigned to the correct one. It is calculated as:

$$\text{Silhouette score} = \frac{s - t}{\max(s, t)} \quad (3.1)$$

where s is the mean of pairwise distances between points of the same cluster (or intra-cluster distance) and t is the mean of pairwise distance of each point of one cluster to another point of the nearest cluster (or nearest cluster distance) [Rou87]. A score of “1” denotes that the clusters are properly separated, whereas a score of “-1” denotes that the clusters are not well separated and that the data points are assigned to the incorrect clusters.

For each case, where the K-means algorithm is carried out, a Silhouette score is computed with a variable number of clusters. The Figure 3.12 shows the Silhouette score with a variable number of clusters of two example images. The optimum number of clusters to utilize for K-means clustering analysis is the number of clusters for which the Silhouette score has the highest value. From Figure 3.12, it can be concluded that this is achieved with either 4 or 5 clusters.

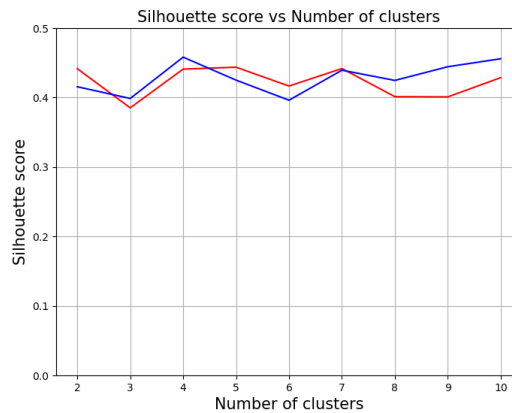


Figure 3.12: Red and blue lines represent the Silhouette score of two different images with various number of clusters. In general Silhouette score differs between 0.4 and 0.5. The best score is achieved with 4 clusters for one image and with 5 clusters for the other, whereas it slightly differs from the score of 4 clusters.

Another approach to find the optimum number of clusters is called the “*Elbow Criterion*” using the “*Sum of Squared Error*” (SSE). The goal of K-means is to organize data in such a way that data similarity within each cluster is maximized while data similarity between clusters is minimized. The distance function is utilized in the cluster as a measure of similarity. As a result, the data similarity is maximized by using the shortest distance from

data points to the centroid point. SSE is the measure for evaluating this data similarity. Via Euclidean distance as the distance function, SSE is calculated as in the following formula:

$$SSE = \sum_{i=0}^n d \quad (3.2)$$

where n is the amount of data that will be clustered and d is the distance between the data and the cluster center according to Nainggolan et al. [NPaST19]. The d is described as follows:

$$d_{ik} = \sqrt{\sum_{j=1}^m (x_{ij} - c_{kj})^2} \quad (3.3)$$

where x_{ij} ($i = 1, \dots, n; j = 1, \dots, m$) with n is the amount of data that will be clustered, m is the number of variables, c the cluster center, k cluster number [NPaST19].

For a better performance of K-means SSE should be minimized. The elbow criterion is a method for determining the best number of clusters by calculating the percentage of the difference between the number of clusters that will form an elbow at a given point. Figure 3.13 shows the SSE values for two different images as an example. For the image in ??, there is a clear elbow point at 2 clusters and an uncertain elbow point at 5 clusters. The image ?? has the elbow points at 2 and 4 clusters.

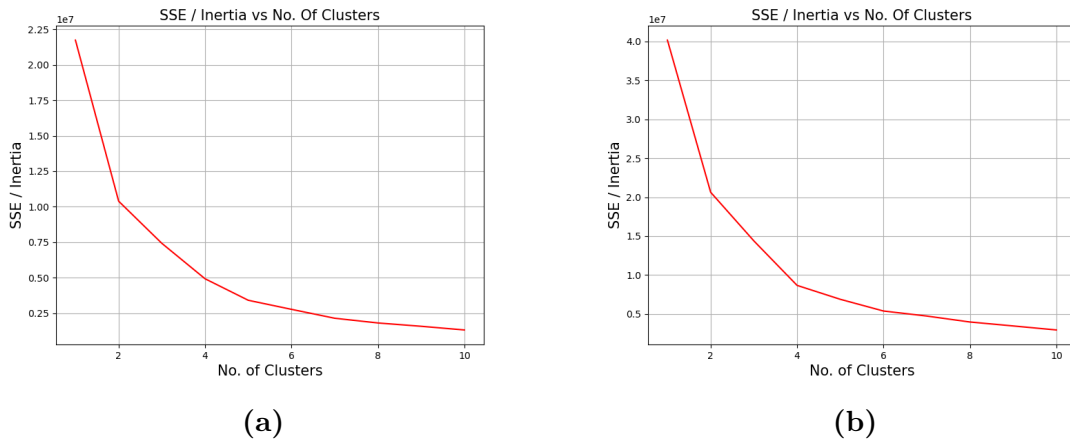


Figure 3.13: Both images show the correlation between SSE value and the number of clusters. The number of clusters is ranging from 1 to 10.

In Figure 3.14, some example images are shown with 2, 4 and 5 clusters. First example image on the left, shows better visual results with five clusters, whereas the optimal cluster number for the right image is two. It is a visual proof that shows the inconsistency of

the optimal number of clusters and to determine this number, other factors should be considered as well.

In the light of SSE plots and Silhouette scores with the whole dataset and the visual evaluation of the formed clusters, the optimal value for the number of clusters in K-means is selected as 4, even though it does not always result in reasonable clustering in our case, depending on how the pore distribution is.

The K-means algorithm is applied after the centroids of contours are determined. The coordinates of centroids are taken as input into K-means analysis and the clusters are identified.

K-means with PCA analysis

The proposed model mentioned in Section 2.3.2 is applied in this method. This is an alternative for clustering pores into pore networks by calculating PCs and applying K-means on the variations of the PCs. In the following, it is explained how the K-means on PCA is conducted.

First step is saving the contour properties in pandas dataframe. This step is similar to the one that is implemented in defect size estimation (Section 3.5). However, it differentiates by the regarded properties. For defect size estimation, the saved properties are the contour center, the contour area, perimeter and the radius of the circumscribed circle. The properties of a contour analyzed in this method are given in the following:

- Aspect ratio ellipse: It is the ratio of major axis to the minor axis of an ellipse.
- Angle ellipse: It gives the orientation of the ellipse.
- Aspect ratio rectangle: It is the ratio of the width to the height of a rectangle.
- Extent rectangle: It is the ratio of the contour area to bounding rectangle area.
- Hu moments: That is a shape feature vector. It can be found on a thresholded image or on the detected contours and describes the shape of the contour. They are invariant to image transformations.
- M Moments: It is also called typically as image moments. Individual objects in a segmented image can be described using moments. The formula for moments of any order is:

$$m_{ij} = \sum_{x,j} (array(x, y)x^i y^j) \quad (3.4)$$

where m_{ij} is the moment for the order (i,j), array calculates the intensity of each pixel (x, y) of the image [Doc17]. This result is called as spatial image moment. In order to ignore the

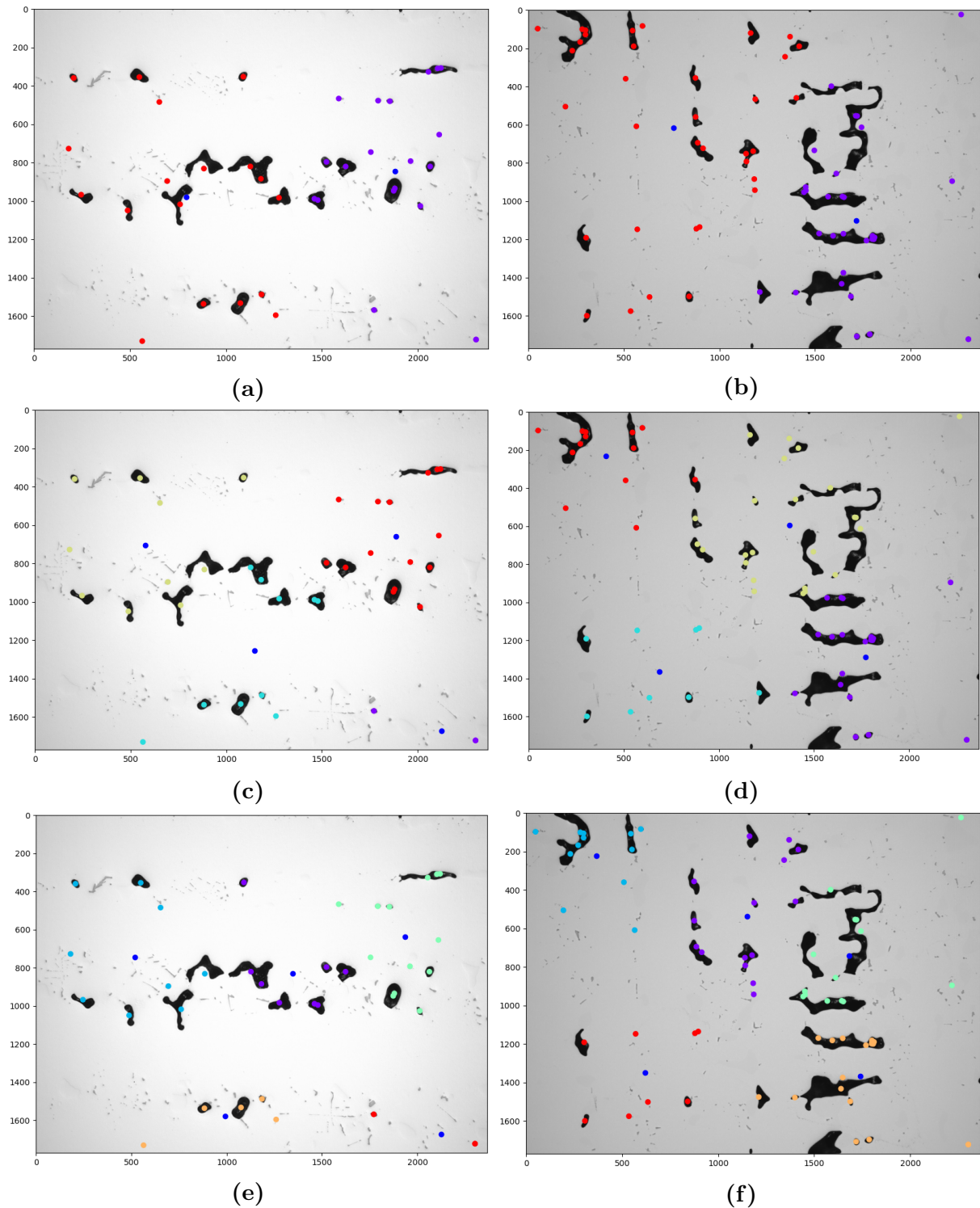


Figure 3.14: The images (a), (b) show the centers of contours assigned into two different clusters represented in red and purple colors. The images (c), (d) and (e), (f) show the contour centers assigned into 4 and 5 clusters, respectively.

effect, where the contours in the image are, one can calculate the central moments:

$$mu_{ij} = \sum_{x,j} (array(x, y)(x - \bar{x})^i(y - \bar{y})^j) \quad (3.5)$$

where the mass center (\bar{x}, \bar{y}) is subtracted from the pixel positions [Doc17]. The mass center is calculated as:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (3.6)$$

The zeroth-order moment describes the contour area, the first-order moment describes the center of mass, the second-order moment describes the moment of inertia according to Singh et al. [SRRL⁺20]. As Rahman et al. showed in their study, the remaining higher-order moments describe projection skewness and kurtosis [RHH19]. Aspect ratios are scale invariant parameters, whereas the moments are scale variant. Both scale variant and scale invariant moments are used in order to determine the better performance. In total 20 properties (10 spatial and 7 central moments, 2 aspect ratios and perimeter) are regarded. Contour area is excluded since the parameter m_{00} refers to the area. Ellipse angle gives not valuable information for the analysis.

The next step is using this input for the PCA analysis. The goal with PCA analysis is projecting the contours from high dimensional matrices onto a new axis. A high dimensional matrix is $n \times p$, where n stands for the number of the detected contours and p is the used contour properties. This is managed by the PCs. The PCs consist of linear combinations of the input properties and are orthogonal to each other. The criteria for selecting the PCs is that the data should be easily distinguishable with the representation through PCs. Each PC is associated with a variance, which represents the ability of parsing the data. Therefore, the PCs with the highest variance should be selected.

Before calculating PCs and the variances, preprocessing of the dataframe is needed. If a contour property value is too small, all moment entity return zero. The zero rows in the dataframe are then removed. Also *NaN* values are removed using the “aspect ratio ellipse”, which is caused by very small contours. Standardisation is also performed so that the variables with higher standard deviation do not cause bigger weight on the PCs. Normalization of the variables makes all variables range from 0 to 1.

Once the PCs are computed via scikit-learn library, the optimal number of PCs is determined by checking the variance that allow the best segregation with the fewest possible PCs. Figure 3.15 shows that a combination of *PC1* and *PC2* would achieve up to 80 % variance, which is sufficient for the analysis. Adding other PCs would not cause a big difference, therefore 2 PCs are selected. K-means clustering is utilized afterwards for the selected PCs.

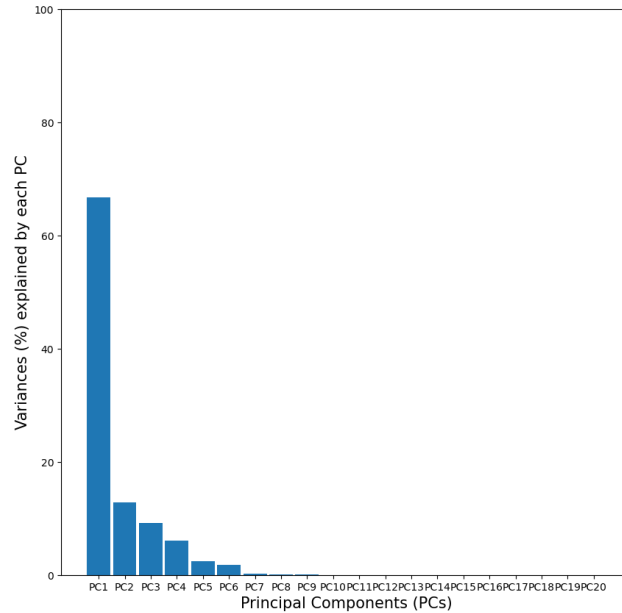


Figure 3.15: Each PC is shown with the variance percentage with bar plot. The sum value of variances results in 100 %. To get the variances, 50 images are checked, 3120 contours are remained after detection and elimination process. Therefore the high dimensional matrix is 3120 x 20.

To better understand the PCs, the correlations between the contour properties in each PC are depicted in Figure 3.16. For instance, the perimeter has a greater impact on the selected two PCs. For the *PC1*, the first- and second-order moments have higher weight on the component than the third-order moments. For *PC20*, m_{20} has a negative weight, hence they have a contrast.

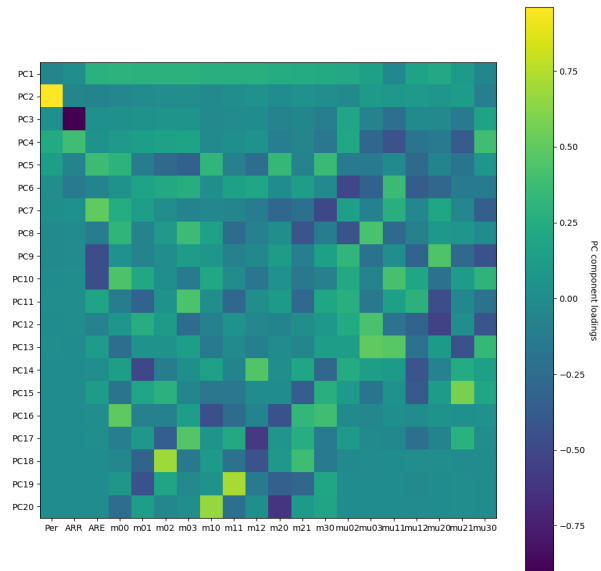


Figure 3.16: The heatmap shows for each PC, the correlations between the original variables. The right-hand bar shows the values that colors refer to. A positive correlation is shown by a positive value, and negative correlation by a negative value.

Applying K-means to PCs is performed the same way as described in Section 3.5. The centroids of each PC is computed and assigned to clusters. To find the optimum cluster amount Silhouette score is used. The 50 test images prove that the optimal value differs between 2 and 5. However, 5 clusters is the majority with the highest score. Therefore, cluster number is chosen as 5. In Figure 3.17, the Silhouette scores of two images are given.

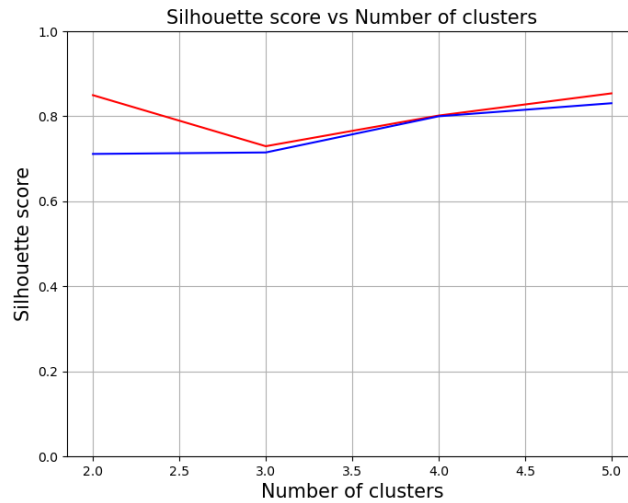


Figure 3.17: Red and blue lines represent the Silhouette score of two different images with $n = 2, 3, 4, 5$ number of clusters. In general, the Silhouette score differs between 0.7 and 0.9. The best score is achieved with 5 clusters for both images.

3.6 Validation

The presented methods are compared with the results of manual estimation. However, the manual estimations are also based on 2D image analysis, which is not capable of giving the 3D insights of the microstructures. For instance, the actual pore size and shape might differ in different 2D image sections. Also the interconnectivity of pores are not clear with one layer images. Therefore, 3D modeling is required to validate the proposed approaches. The automated serial sectioning method presented in Section 2.4 from Robo-Met is used for the validation task. For the analysis, appropriate sample materials were picked out and sent to Robo-Met for 3D rendering. This method allows an optical magnification of 50x (approx. $2 \mu\text{m} \times y$ resolution), and a section pitch of approx. $5 \mu\text{m}$.

Material selection

Three samples and one spare sample of irregular shapes were chosen for this study. There are various criteria used to pick these materials in order to ensure that these contain a large number of pore networks to investigate. For instance, they are all picked from *polycrystalline*² superalloys, a casting process that results in a higher porosity. Furthermore,

²Single crystal superalloys are produced through directional solidification techniques whereby the final component comprises only of a single grain, whereas polycrystals are composed of many individual grains. Single crystals have a better creep resistance and polycrystals have higher tensile strength [LWS⁺19].

they are not processed through *Hot Isostatic Pressing* (HIP) during the heat treatment process, since HIP reduces the porosity inside the material. After checking the supply, some candidates were picked and the *computed tomography* (ct) images were checked if they have enough porosity. Final selection for the materials were for two superalloys: *INCONEL 718* (IN-718) and *INCONEL 713* (IN-713), which are typically used in aircraft engine components. Two samples from IN-718 (Sample 1 and 2) and two samples from IN-713 (Sample 3 and as a spare Sample 4) were cut. Cuts were extracted from areas of turbine components where ct imaging had shown a high likelihood of porosity. Sample 1 and 2 are approx. $20 \times 10 \times 4 \mu\text{m}^3$. Sample 5 is approx. $20 \times 5 \times 2 \mu\text{m}^3$.

Comparison methods 2D vs. 3D

To find how realistic the 2D results, a comparison of 2D analysis with 3D results is needed. For that purpose, different methods are used. The first parameter to be compared is the porosity percentage. Porosity percentage depicts the ratio of pores to the whole volume (in 3D) or area (in 2D) of the material. The value that is found in 2D image layers are compared to the porosity percentage value of pores in 3D analysis.

For single pore size comparison, *Equivalent Spherical Diameter* (ESD) is applied. It is calculated with the following formula (taken from the study of Pabst et al.[PG07]):

$$D_{volume} = \left(\frac{6}{\pi} V_{particle}\right)^{1/3} \quad (3.7)$$

where D_{volume} is the diameter of a sphere with the same volume as the particle $V_{particle}$, i.e. volume-equivalent sphere diameter. ESD for a cross sectional area in 2D is calculated as:

$$D_{surface} = \left(\frac{6}{\pi} S_{particle}\right)^{1/2} \quad (3.8)$$

where $D_{surface}$ is the surface-equivalent sphere diameter, i.e. the diameter of a sphere with the same surface as the particle $S_{particle}$ [PG07].

Furthermore, the key part of the comparison is determining the appropriate threshold for accepting pores in a pore network. For that purpose, a crop area is selected and the 2D image layer stack (100 layers) of this area is processed with the ‘‘Fiji-ImageJ’’ image tool. The program returns the interconnected pores with the same color in the 2D images as shown in Figure 3.18). These images are investigated for validating the used threshold methods mentioned in Section 2.2.

In order to perform the comparison, the contours are identified which are the same color. Since images are compressed, the RGB values are not exactly the same for each pixel that belongs to the same color. Therefore, the RGB images are first converted to HSV – H is for hue, S stands for saturation and V represents value i.e. brightness – to get the same colors in a smaller interval. Then an interval of 10 for the H value is used to identify

the same colors according to some test applications. After the same groups are identified, the majority color group in the biggest cluster group is determined. Consequently, it is checked, whether this cluster contains all the pores with the same color on the image.

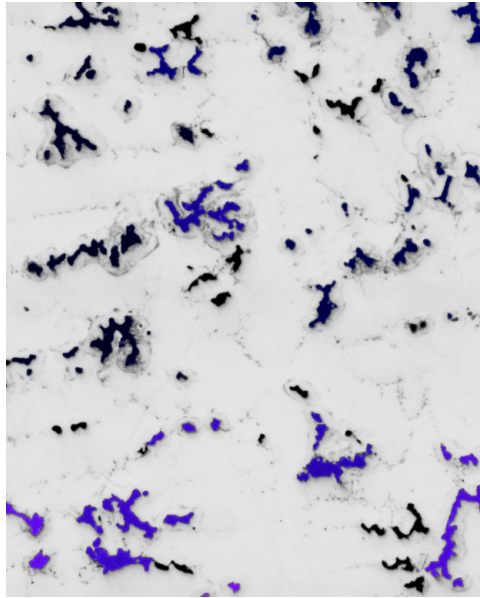


Figure 3.18: This image is obtained by the image tool Fiji. The same colors depict the same pores. To create this image, a maximum image stack of 2GB can be used in Fiji and this process takes approx. 20 hours.

Chapter 4

Evaluation

4.1 Evaluation metrics

This section presents the metrics used for evaluating the results of the different methods and their performance. Through these metrics, it is possible to quantify the error of a given measurement. An error can be defined as the difference between measurement's outcome and the measurand's true value. For instance, the measurand's true value is the manual measurement of the defect size and the measurement's outcome is the algorithm's solution in the comparison of methods with manual approach. There are four different categories to describe the performance of a prediction, where the result is binary, i.e. the outcome can be referred as positive or negative:

- *True positive (TP)* : The result where the model predicts the positive class correctly.
- *True negative (TN)* : The result where the model predicts the negative class correctly.
- *False positive (FP)* : The result where the model predicts the negative class incorrectly as positive.
- *False negative (FN)* : The result where the model predicts the positive class incorrectly as negative [HPJ⁺21].

An example of such a prediction is defining the biggest pore cluster on an image with different methods. The positive value is regarded as the pores that belong to the biggest cluster. The negative value represent the pores, which do not contribute to the biggest cluster. The *TP* becomes the pores that are assigned as part of the biggest cluster, which is also true in "reality". The pores that are actually not in that cluster but assigned as positive build the *FP* group. Compared to *TP* and *FP*, *TN* and *FN* are the metrics for the pores that are assigned as negative. The *TN* describes the a correct assignment, whereas *FN* defines the group that should be in biggest cluster. These values give an insight on the results with the help of different measures such as "*Accuracy*", "*Recall*" and "*Precision*",

which are described as in the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

Accuracy emphasizes overall performance of the model predictions. High accuracy equates to more accurate predictions. Precision highlights the impact of FP results. Higher precision is a result of a remarkable accuracy, even if the negative results are dominating on the image. Higher recall ensures that the major part of the positive results are predicted correctly.

To compare the defect sizes obtained with various algorithms, the arithmetical mean value \bar{x} and standard variation s are used. The mean value gives an insight about where approximately the data points lie. In statistics, the standard deviation is a measure of the amount of variation or dispersion of a set of values [BA96]. A small standard deviation value implies that the values are close to the set's mean, whereas a large standard deviation value shows that the values are spread out over a larger range. The values s and \bar{x} are calculated as:

$$\bar{x} = \sum_{i=1}^n \frac{x_i}{n} \quad (4.4)$$

$$s = \left(\frac{(\sum_{i=1}^n x_i - \bar{x})^2}{n - 1} \right)^{1/2} \quad (4.5)$$

where n is the number of measurements and x_i are the results for defect sizes. The last operation of standard deviation is taking the square of the term in the parenthesis. That term builds the *variance* ($VAR(X)$) of the dataset, which also explains the dispersion of the data. In comparison to standard deviation, variance has the drawback that its units from random variables differ and thus, standard deviation is more commonly used.

4.2 Results

In this section, the results of the methods that are presented in the previous sections are shown. Additionally, the evaluation metrics are used to draw conclusions. The results are grouped by their relevance.

Image type estimation

The image type estimation algorithm forms the beginning of the pipeline and therefore its accuracy is important for the rest of the analysis. During implementation its accuracy was already tested with company images and it was tested again with an extended dataset using the images from Robo-Met. The achieved accuracy of 75 pore nest images was 96 % and of 65 overview images 85%. The accuracy of the entire dataset achieved up to 90%. Two example images which were not assigned correctly are shown in Figure 4.1.

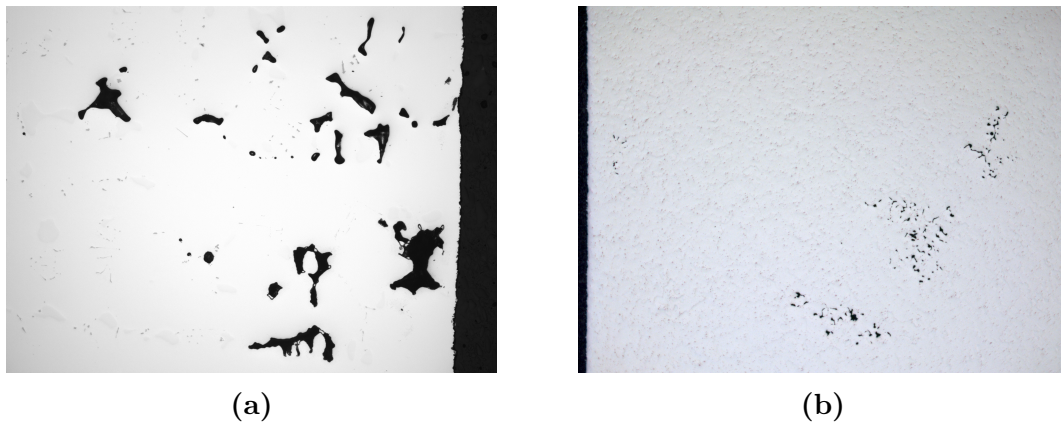


Figure 4.1: The image (a) shows a pore nest image, which was assigned to the overview class. The overview image (b) was detected as pore nest image.

Exploratory analysis

Before calculating the defect sizes, an exploratory analysis was executed in order to have better insights on the dataset and its microstructural features. In Figure 4.2, the manually estimated defect sizes in pore nest images and overview images are given.

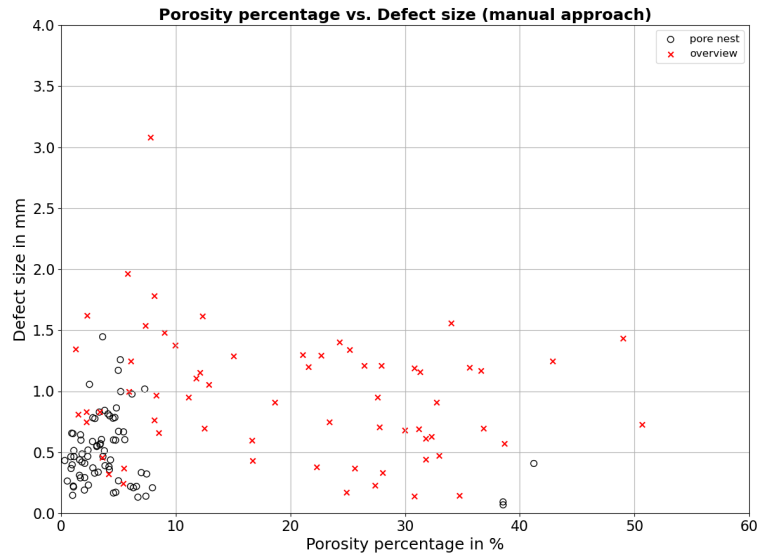


Figure 4.2: The image shows for 75 pore nest and 65 overview images the manually estimated defect sizes. Red points represent overview images and black circles represent pore nest images.

The mean value of the defect sizes is $770 \mu\text{m}$ and the standard deviation is $454 \mu\text{m}$. There is no direct correlation between porosity percentage and the defect size value. The pore nest images have overall lower porosity than the overview images. Their mean defect size value is $512 \mu\text{m}$ with a standard variation of $286 \mu\text{m}$, whereas as the mean defect size from overview images is $950 \mu\text{m}$ with a standard variation of $510 \mu\text{m}$.

In a similar manner, increasing porosity percentage does not influence the maximum values of perimeter and max. Feret diameter on an image. However, it results in a slightly increased mean value of max. Feret and perimeter (see Figure 4.3 and 4.4). The square root area of a pore has a variance of 18%, whereas the Feret diameter has a variance of 0.4%.

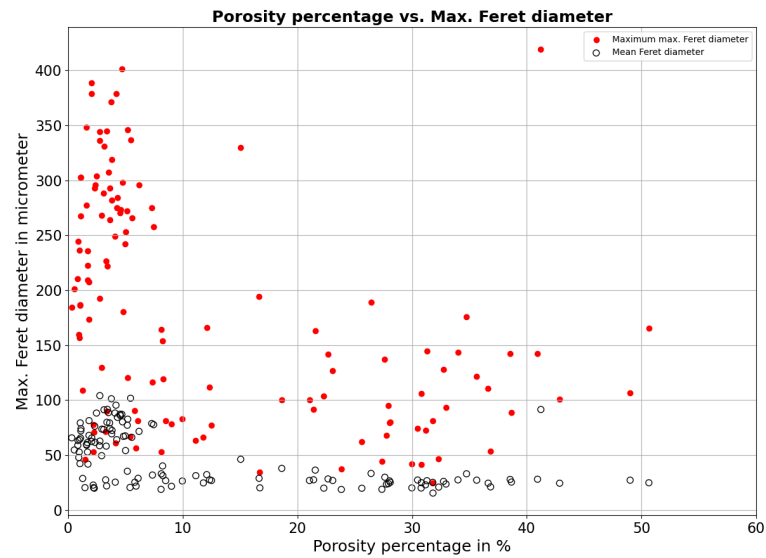


Figure 4.3: This plot shows max. Feret diameters of the detected contours in 140 component images. Black circles represent the mean value of max. Feret diameters in each image, whereas the red points represent the maximum value. They are depicted in relation to the porosity percentage.

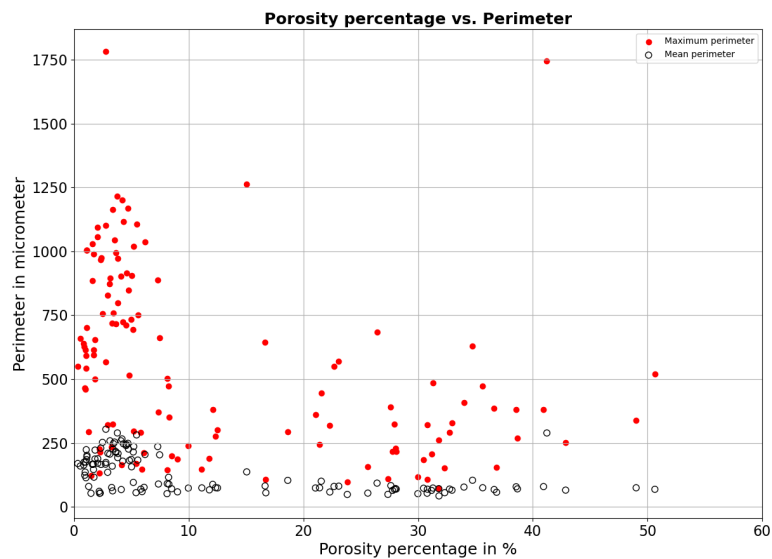


Figure 4.4: This plot shows perimeters of the detected contours in 140 component images. Black circles represent the mean value of the perimeters in each image, whereas the red points represent the maximum perimeter value. They are depicted in relation to the porosity percentage.

Defect size estimation

Manual check results build the ground truth for comparing various defect size estimation methods. For that purpose, the defect sizes of the dataset, which were estimated manually were given in Figure 4.2.

Figure 4.5 shows the bounding rectangles with different thresholds on an example image. There, it is also possible to see the different sizes of bounding rectangles.

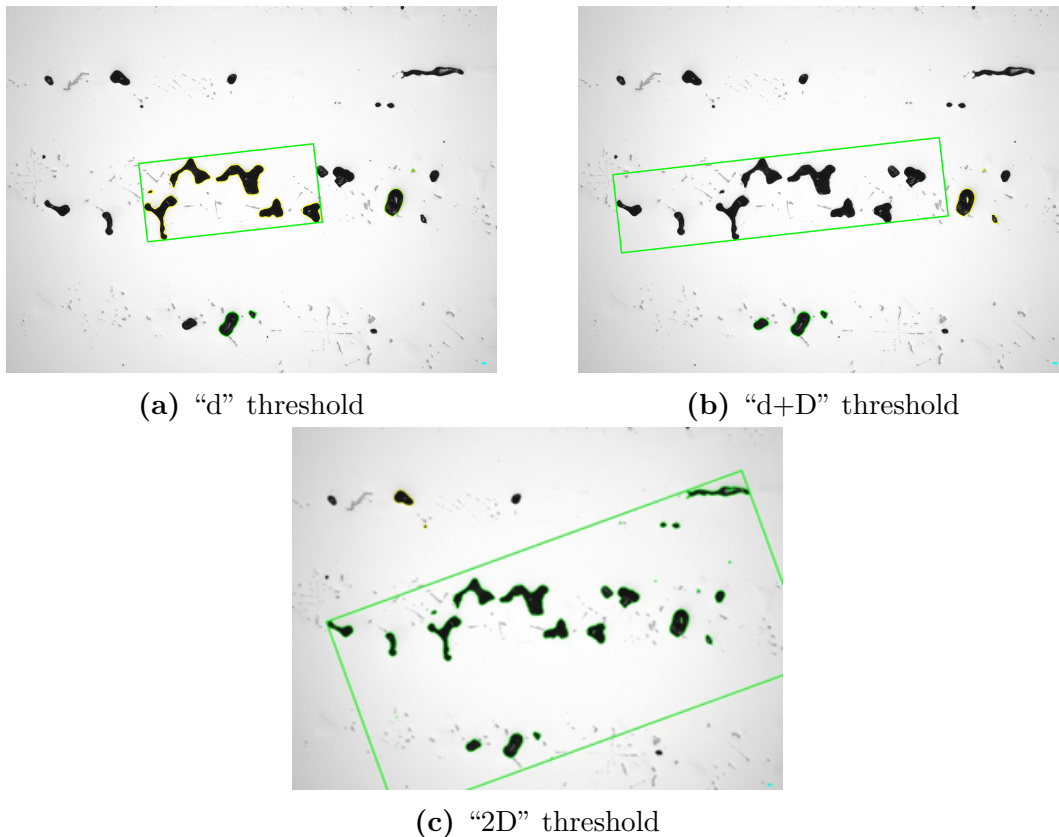
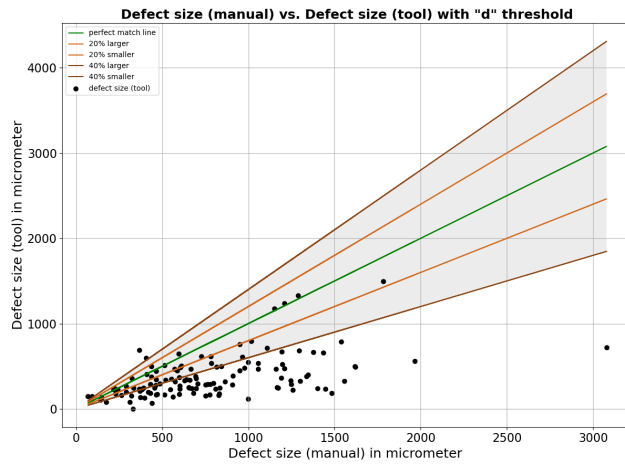
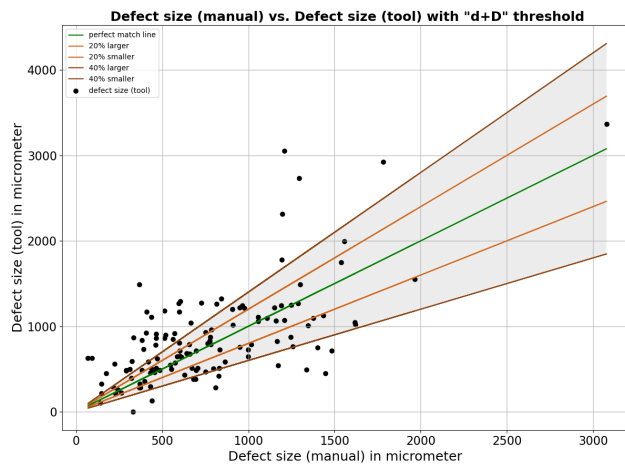


Figure 4.5: First image on the left-hand side shows the bounding rectangle around the detected pores with "d" threshold. Second image shows with "d+D" and third image shows with "2D" accordingly.

Figure 4.6 shows the defect sizes calculated with bounding rectangle algorithm with different minimum criteria for pore network acceptance and the defect sizes estimated manually. It shows that the criteria "2D" results in bigger defect size and "d" results in the smallest defect size. Furthermore, it can be seen that manual estimations are more close to "d+D" method when one compares the data points in the 40% and 20% error intervals. The "d+D" threshold has around 10% error in average, where the "2D" has 30% and "d" has 45%. The average time for estimating the defect size on one image with bounding rectangle algorithm was 55 seconds.



(a) “d” threshold



(b) “d+D” threshold



(c) “2D” threshold

Figure 4.6: This plot shows the defect sizes estimated with different thresholds in comparison with manually estimated defect sizes. The green line is perfect match line where the light and dark brown colors depict 20% and 40% error lines, accordingly.

Figure 4.7 shows the defect size estimated with pore area model. It is given with manual estimation for a better overview. It is possible to see that the defect sizes with pore area model are smaller than manual estimations. The mean estimated defect size is $180\mu\text{m}$ with the standard deviation $102\mu\text{m}$. The presented defect sizes are estimated with the contour group estimated with “2D” threshold. Since this threshold produces the largest defect size of all the thresholds, the “d” and “d+D” thresholds would produce significantly smaller defect sizes. Therefore, it is not necessary to check the pore area model with other thresholds.

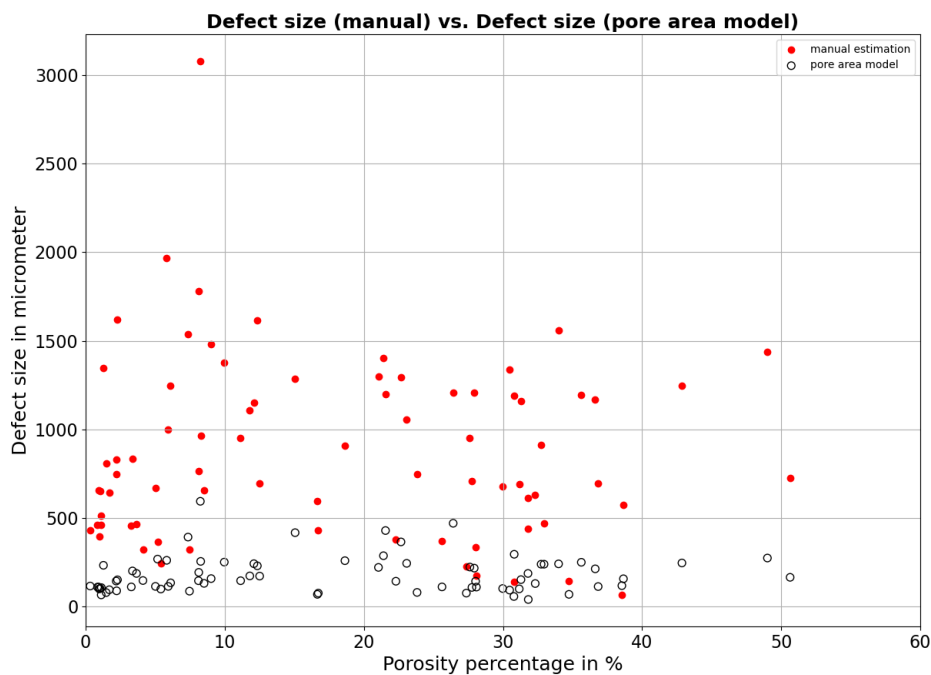


Figure 4.7: This plot shows defect sizes of 140 component images. Red points represent the manual estimation results, whereas black circles points represent the defect sizes calculated with pore area model. They are depicted in relation to the porosity percentage of the image.

Another method to estimate defect size is using the Hough transformation. Figure 4.8 shows two example images which were applied the Hough transformation and the longest detected line was drawn. The analysis for one image was about 6 seconds on average. One can conclude that the problem with overview images is detection of the outer contour of the component as a line. Therefore, the analysis does not provide the true defect value. However, on a pore nest image, the results were very similar to the manual estimations. Figure 4.9 presents defect sizes estimated manually. With a mean value of $2489\mu\text{m}$, Hough transformation reached larger values than manual results, which were caused by the issue in overview images.

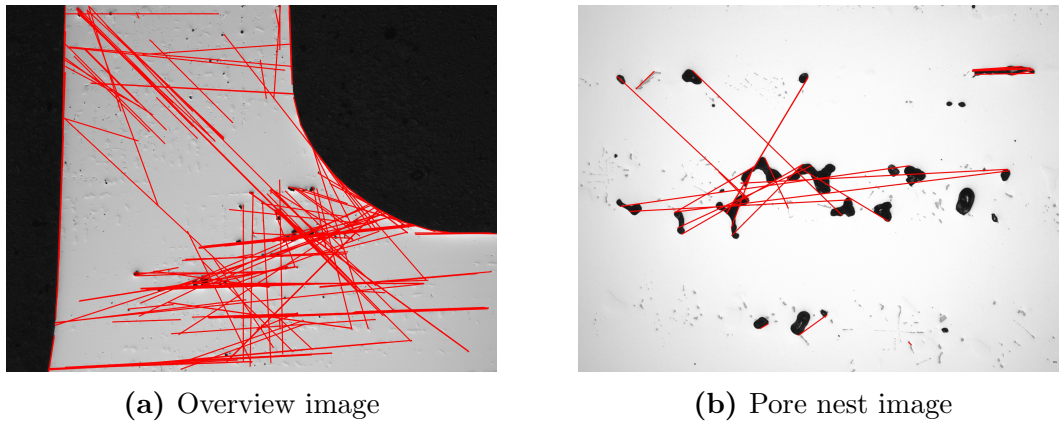


Figure 4.8: The figure (a) shows an overview image with the longest detected line of $4625 \mu\text{m}$ by the Hough transformation algorithm. The image(b) shows a pore nest image with the longest detected line of $1038 \mu\text{m}$. The red lines represent the detected lines with minimum line length of 10 pixels and maximum line gap of 250 pixels.

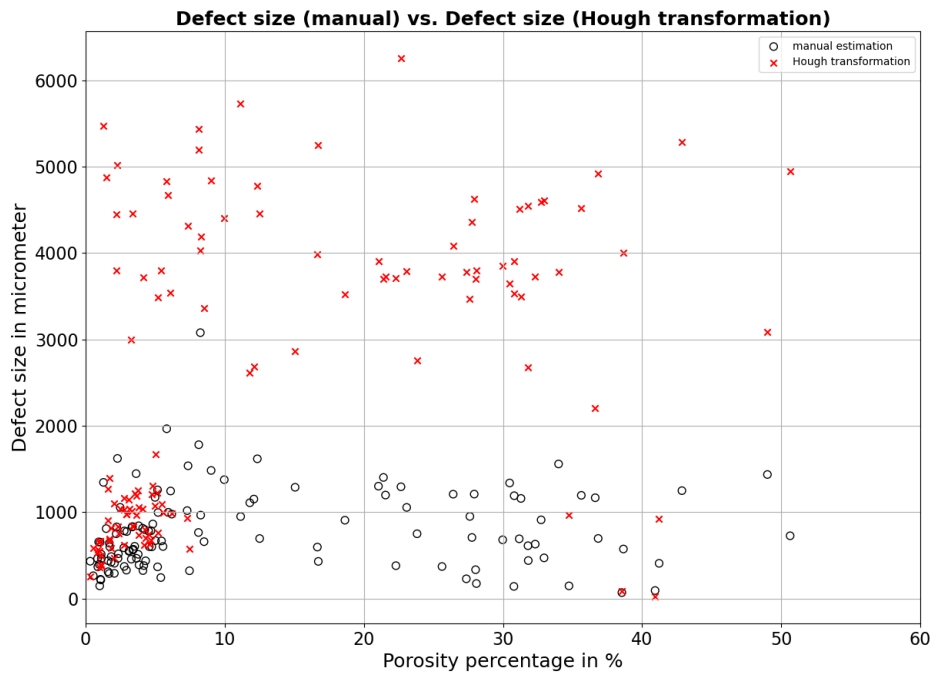
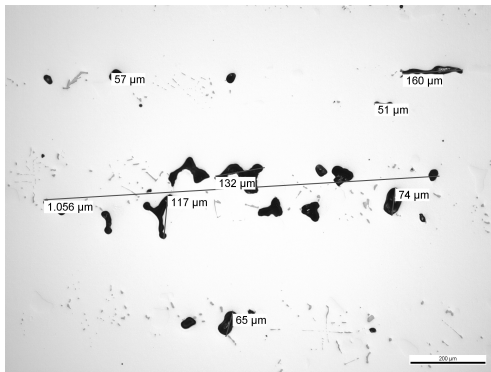


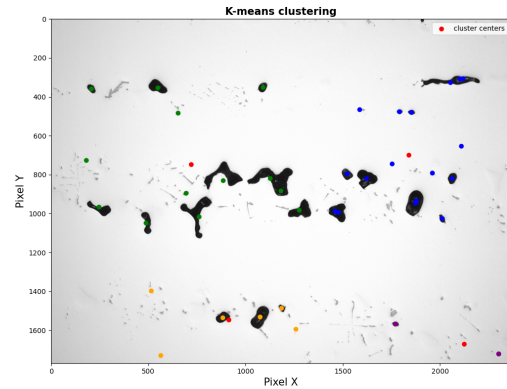
Figure 4.9: This plot shows defect sizes of 140 component images. Black circles represent the manual estimation results, whereas red points represent the defect sizes calculated with Hough transformation. They are depicted in relation to the porosity percentage of the image.

Two examples with K-means algorithm are shown in Figure 4.10. The average time for

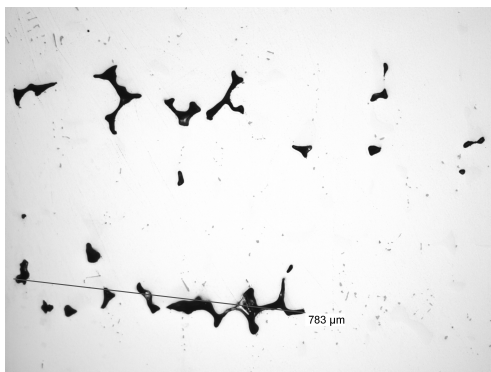
analyzing one image was 8 seconds. Visual check proved that the clusters do not match the manually detected groups. The K-means method resulted in smaller pore groups than manual estimation. Therefore, the defect sizes estimated by K-means algorithm are calculated in combination with PCA analysis. The results with two example images are to be found in Figure 4.11.



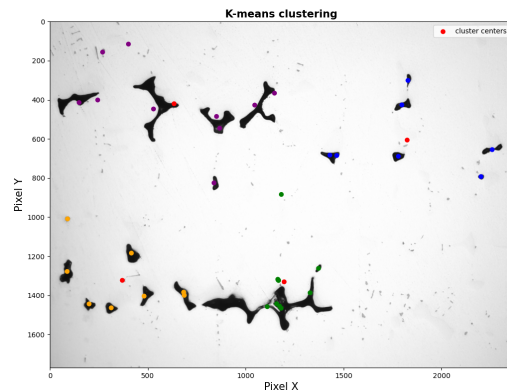
(a) Manual estimation



(b) K-means clustering with 4 clusters



(c) Manual estimation



(d) K-means clustering with 4 clusters

Figure 4.10: The images on the left show the manually checked images. On the image (a), the biggest cluster has the size $1056\mu\text{m}$ and the image (c) has the value of $783\mu\text{m}$. The images (b) and (c) show the K-means algorithm applied to the centroids of the detected contours. The red points represent the four cluster centers and other colors represent different clusters, respectively.

The two examples in Figure 4.11 illustrates that this algorithm takes the form of the pores into account and therefore the distances between the pores are ignored.

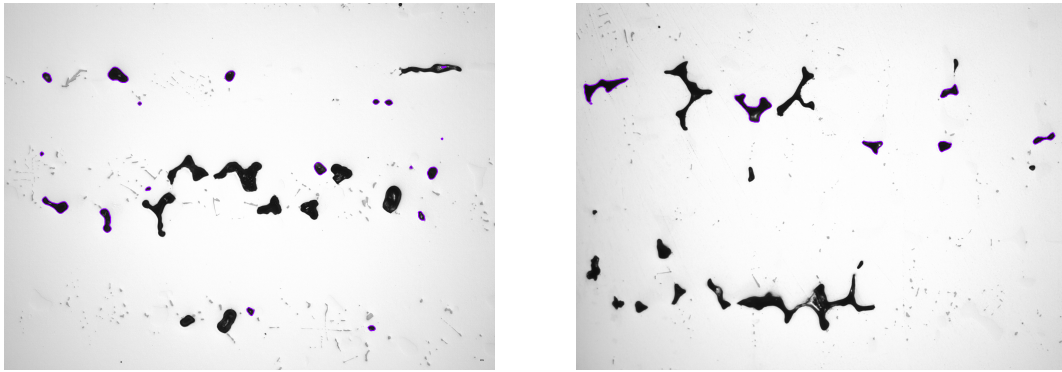


Figure 4.11: PCA analysis with K-means algorithm are applied to both images. The contours marked with purple lines are the biggest cluster resulted from the algorithm.

3D comparison

The sample analysis and 3D rendering took two months to complete. The magnification achieved after rerrendering was 50x. Some data was lost due to planarization at the beginning and at the end. For each sample the final parameters after the analysis are as follows:

- Sample 1: A section pitch of average $5.6 \mu\text{m}$ was achieved. 625 slices were used for the analysis. A depth of 3500 microns was analyzed. The voxel resolution was $21 \mu\text{m}^3$. The 3D rendered image is presented in Figure 4.12.

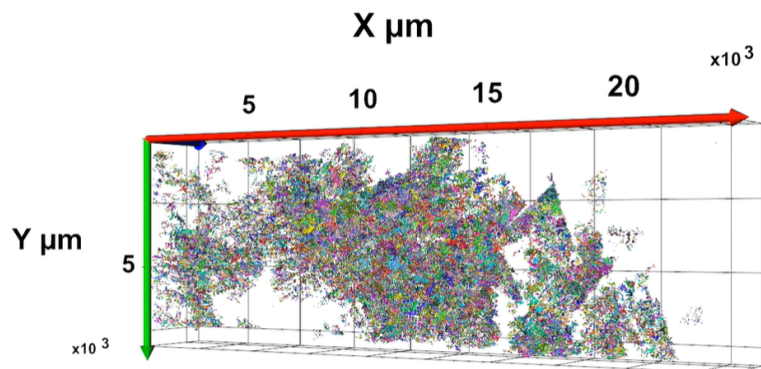


Figure 4.12: This representation is obtained with $20 \times 8 \times 2$ subsampling. The illustrated pores have a minimum volume of $10.000 \mu\text{m}^3$.

- Sample 2: A section pitch of average $5.5 \mu\text{m}$ was achieved. 643 slices (28 GB image stack) was used for the analysis. A depth of 3500 microns was analyzed. The voxel resolution was $20.7 \mu\text{m}^3$. The 3D rendered image is presented in Figure 4.13.

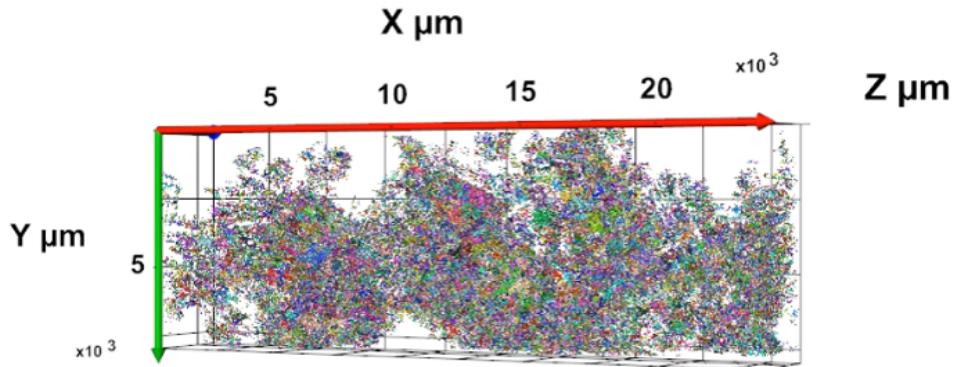


Figure 4.13: This representation is obtained with $20 \times 8 \times 2$ subsampling. The illustrated pores have the minimum volume of $100 \mu\text{m}^3$.

- Sample 3: A section pitch of average $4.7 \mu\text{m}$ was achieved. 407 slices were used for the analysis. A depth of 1950 microns was analyzed. The voxel resolution was $24.3 \mu\text{m}^3$. The 3D rendered image is presented in Figure 4.14.

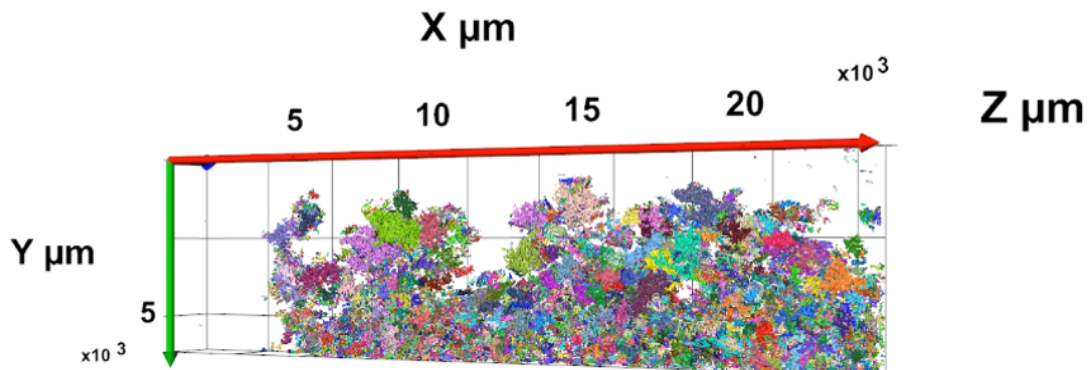


Figure 4.14: This representation is obtained with $20 \times 5 \times 1$ subsampling. The illustrated pores have the minimum volume of $10.000 \mu\text{m}^3$.

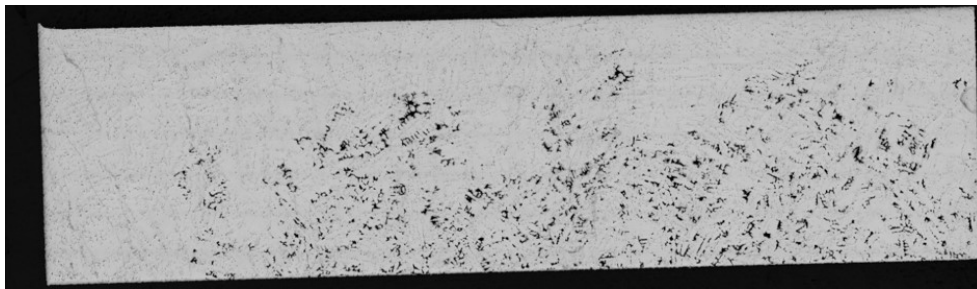
The statistical analysis is carried out on Sample 3, because the other two samples appear to have extremely dense porosity, that performing the 2D analysis does not bring meaningful conclusions. The results of the microstructural features for the Sample 3 are shown in Table 4.1. The mean value for area fraction was 3.38%, where the mean volume fraction was with 1.83% around the half value of the area fraction. The mean max. Feret and area/volume mean values depict how big a single pore in reality is and how it looks like in

Table 4.1: Microstructural analysis 2D vs. 3D

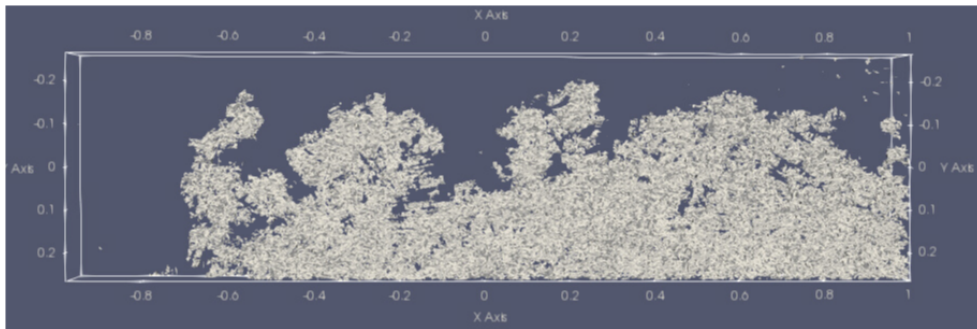
	Area/Volume Fraction	Max Feret (mean)	Max Feret (std)	Area/Volume (mean)	Area/Volume (std)	ESD
2D	3.38 %	43 μm	0.56 μm	639 μm^2	21 μm^2	35 μm
3D	1.83 %	199 μm	115 μm	198870 μm^3	250000 μm^3	72 μm

2D image. The ESD value (shown in Section 3.6) of 2D is around half of the ESD of 3D analysis.

In Figure 4.15, a single layer of Sample 3 is compared to the 3D porosity only rendering. Looking at the figure it is clear that, not all pores are visible in every layer.



(a) 2D image of the first layer from Sample 3



(b) 3D model of the Sample 3 obtained with Paraview software

Figure 4.15: The figure presents one of the layer in 2D (a) with 3D model of the sample (b). The microstructures are depicted as 3D surface mesh of triangles with Paraview software.

For finding the reasonable distance threshold, the colored pore groups are analyzed as shown in Figure 3.18. The performance of the three threshold settings is examined using two metrics: recall and precision. Recall is 100%, when there is no outlier in detection. Outlier implies the pores that have the same color but not assigned to the same contour group. Precision emphasizes the over estimation of pores. If there are many other pores which have various colors than the dominating one in the detected pore group, the precision is low. The best performance is considered to be a threshold setting with high recall and precision. In Table 4.2, the results of the distance thresholds are given.

Table 4.2: Performance of different threshold settings

	Recall	Precision
2D	85%	41%
d+D	80%	85%
d	13%	45%

The Table 4.2 demonstrates the highest recall with “2D” threshold, however the precision is significantly lower than “d+D” setting. The threshold value “d” does not achieve high recall or high precision.

4.3 Discussion

In this section, the introduced results are discussed and evaluated. The evaluation is grouped by the same structure as in the previous section.

Image type estimation

The results show that image type estimation algorithm performs well (with 90% accuracy). Checking the incorrectly assigned images, the wrong estimation of image type does not result in wrong consequences due to their “unexpected” black and white pixel ratio. If an image is assigned as an overview image, it means that it requires elimination of outer contour and different method in percentage calculation, so that the background is not taken into account. It is exactly the case in pore nest images if they are detected as overview images. For instance they have also part of a background on the image. Hence, that image also requires the steps as in overview images. In the end, the importance of image type estimation lies in different preprocessing steps. The used methods therefore serves the aim of applying the right methods and can be used.

Exploratory analysis

The first outcome of exploratory analysis is that, the porosity percentage does not allow to draw conclusions about the defect size, since there is no direct correlation between them. Also pore network images in general show smaller defect size than overview images, since the regarded area is restricted. The size of overview images differ from each other and it might be possible to get a better overview of the component. However, the analysis with pore nest images are more stable. It highlights the need of having the overview images in the right order wherever one can estimate where exactly in the component they are from.

Square root area of a pore has a larger variance than the max. Feret diameter (see Section 4.2). Based on this result, the size of a pore is better defined by with its max. Feret diameter instead of its square root area.

Defect size estimation

The comparison of defect sizes estimated manually and with bounding rectangle using three different thresholds showed that “d” and “d+D” distance thresholds result in more close values to manual measurements than “2D” threshold. The Figure 4.16 highlights the overestimation of defect sizes with “2D” threshold and the underestimation with “d” threshold. Therefore, the threshold “d+D” can be considered as the actual used threshold while performing manual defect size estimations. However, this needs to be validated.

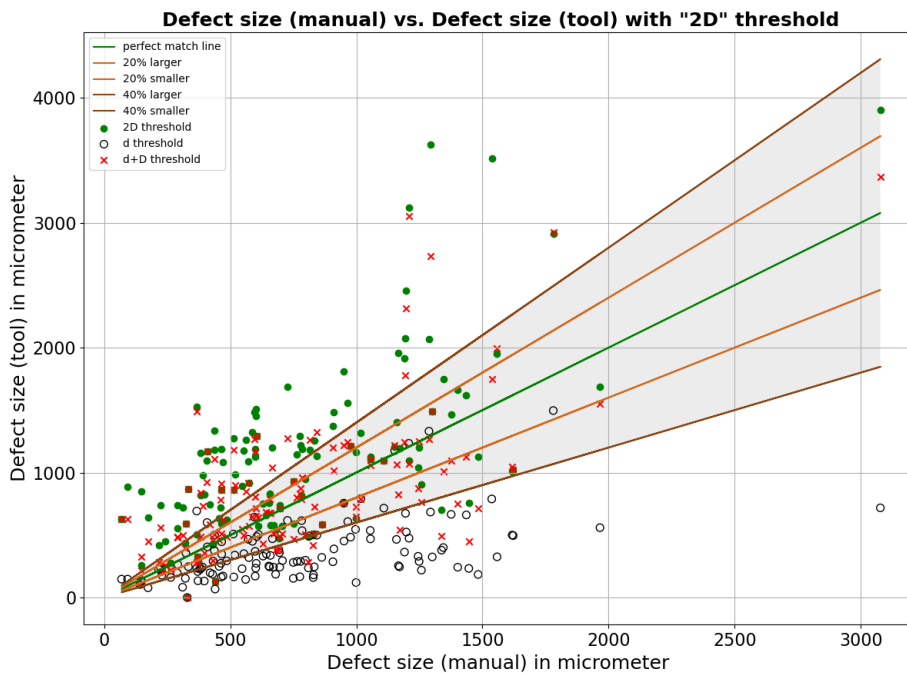


Figure 4.16: This plot shows the defect sizes estimated with different thresholds in comparison with manually estimated defect sizes. The green line is perfect match line where the light and dark brown colors depict 20% and 40% error lines, accordingly.

For the challenge with the Hough transformation detecting outer contours, one solution might be setting a maximum value for line. For a better performance the gap size between pixels in order to be detected as line can also be optimized. On the other hand, it is a

faster method than bounding rectangle, since it eliminates the need to examine all of the distances between contours.

The advantage of the K-means algorithm and PCA analysis is that these methods do not necessitate a lot of training time or manually labeled data for the analyses. Furthermore, there is an imbalanced class problem where the number of pores in the biggest cluster is to other pores usually lower than other pores, using supervised machine learning techniques such as random forests in such a case would not detect the biggest cluster with high accuracy.

The issue with the K-means algorithm is that it separates the biggest cluster into multiple smaller ones when there are not many pore groups on the image. The detection on images with 4 or more pore groups performs well. Therefore, one additional step can be added, where after defining the pore groups checking the distances between groups and unify two groups into one, if they are below a set threshold (e.g. “d”, “d+D”, “2D”).

K-means algorithm with PCA analysis is in some of the cases profitable, however for the example images in Figure 4.11, the marked pores are not similar in the group that are manually estimated. Furthermore, if one calculates the defect size accordingly, the result is higher than expected values, since the distribution of the pores are dispersed.

The best performance in computation time was achieved by Hough transformation. K-means clustering is also faster than threshold distance algorithm, however its total duration depends on the defect size estimation method. Although the bounding rectangle method is the slowest, it is adequate when compared to manual processing time.

3D comparison

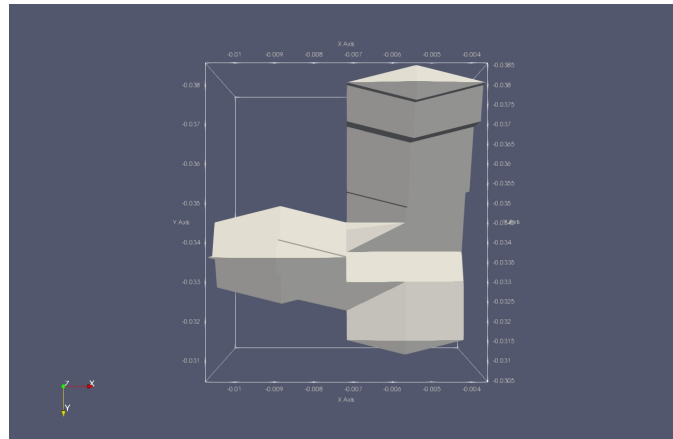
The 3D models are particularly beneficial for understanding the interconnection of pores. This becomes clear in the Figure 4.19. There, the different appearance of the same pore groups becomes visible.

The porosity percentage in 2D is around double of the porosity percentage in 3D (see Table 4.3). This may lead to the classification of pictures with low porosity as important. However, the high amount of pores implies that the 2D analysis is not sufficient for checking the pores.

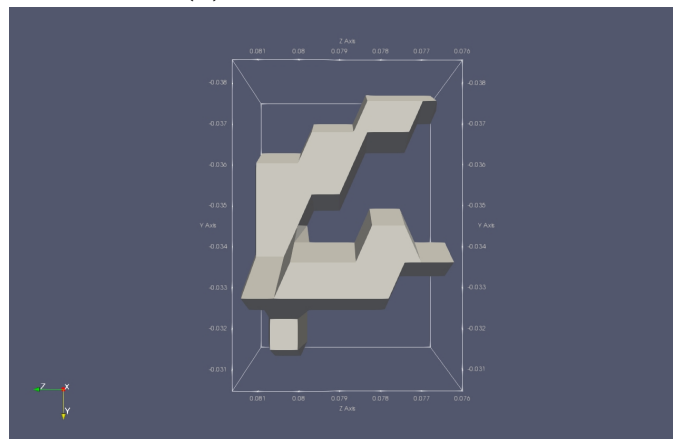
The ESD values are also different from 2D and 3D analysis. 3D analysis shows larger values than 2D results (see Table 4.3). One reason for this could be the fact that pores do not always have a spherical shape (see Figure 4.17). Hence, the ESD may be calculated differently from a 2D image, in order to get a more realistic result.

Table 4.3: Microstructural analysis 2D vs. 3D

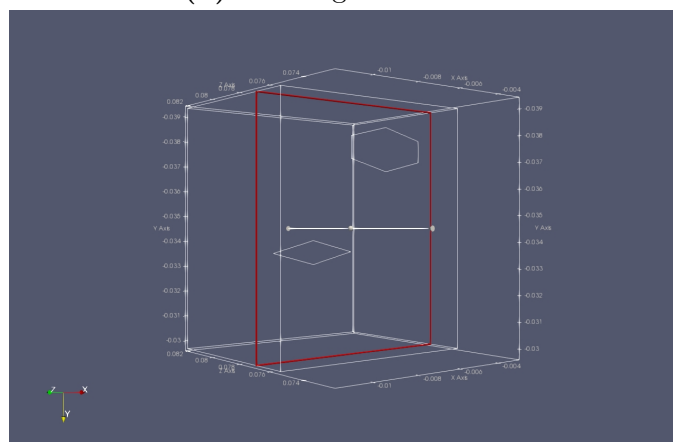
	Area/Volume Fraction	ESD
2D	3.38 %	35 μm
3D	1.83 %	72 μm



(a) 3D image in XY Axis



(b) 3D image in ZY Axis



(c) 2D image in XYZ Axis

Figure 4.17: An example of a single pore (a) in XY axis, (b) in ZY axis and (c) shows one slice of this pore where all axis can be seen. The pore is selected from Sample 3.

The distance threshold evaluation for pore groups showed the best results with “d+D” threshold. It has a similar recall as “2D” and a higher precision percentage. Even though, the first visual checks showed also good results with “d” algorithm, its recall including the same pores in bounding rectangular was not high. Moreover, the precision was low, because the “d” threshold only detects a small number of contours, and the error has a larger impact on the precision (see Figure 4.18).

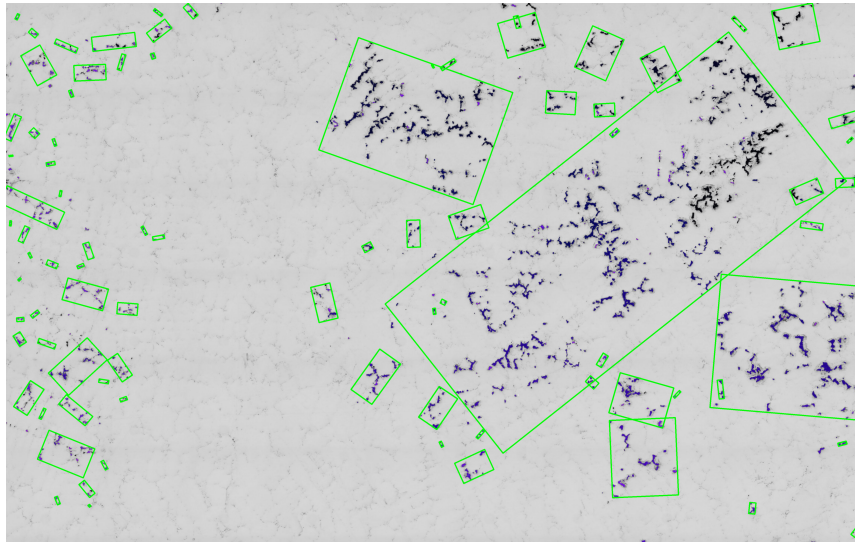


Figure 4.18: This plot shows an example image with “d” threshold estimated bounding rectangles.

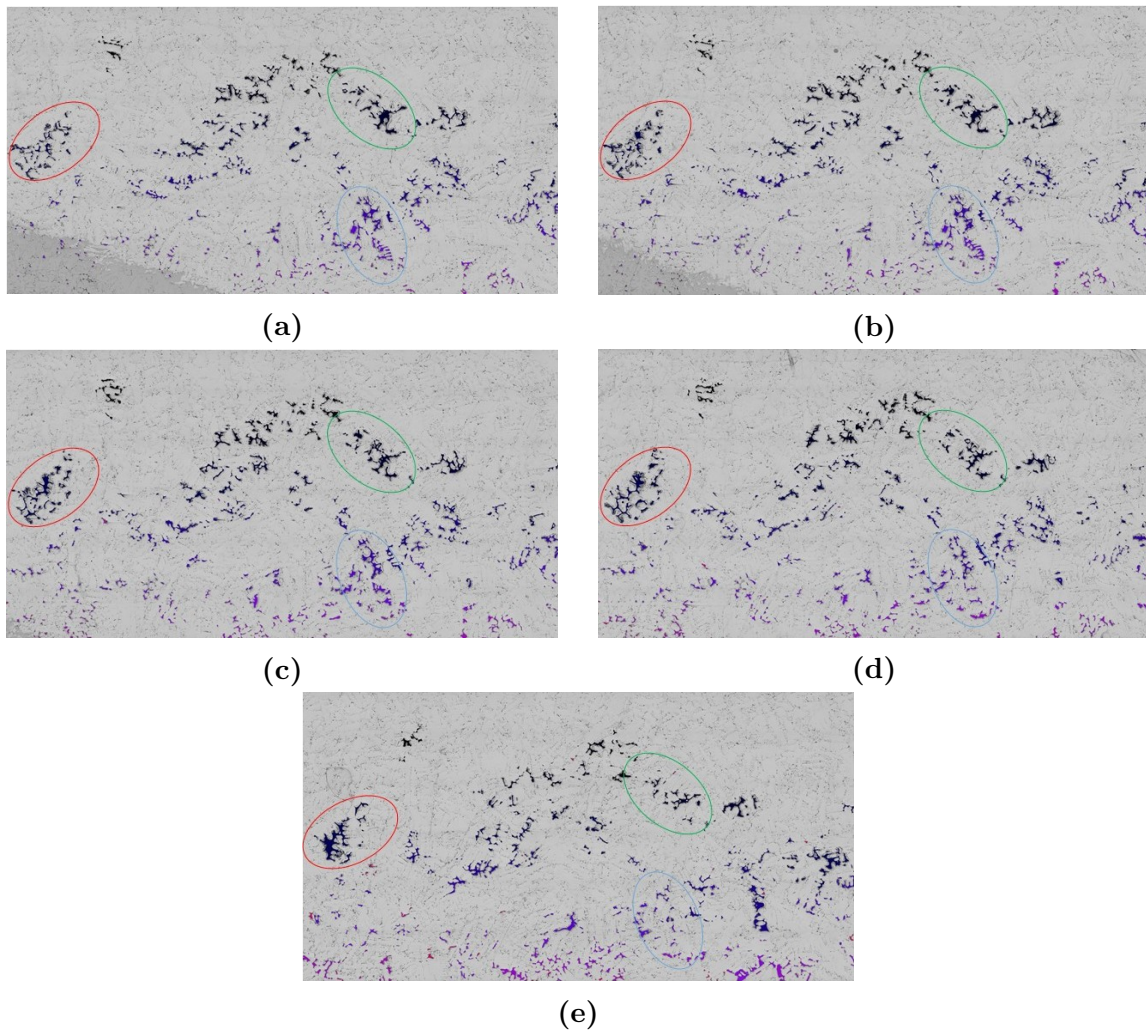


Figure 4.19: The images (a)-(e) show different layers of Sample 3. The red, green and blue marked areas highlight the same pore groups, which appear differently on different layers.

Chapter 5

Conclusion

Casting defects have a big influence on the mechanical properties of the thermally and mechanically highly stressed components. Thus, the evaluation of porosity in turbine components made of nickel-based superalloys is a key criteria in quality assessment. Therefore, it is necessary to detect and evaluate the defect sizes in materials and to analyze the morphology of casting pores.

Metallographic examination is performed to determine the defect size and to understand the critical pictures. To begin with the first challenge, manual examination results are not reproducible. Secondly, monitoring for the threshold distance between pores is not always precise. So there is no strict rule in defining the defect size. Moreover, the checks are carried out in two dimensional (2D) images and thus the plausibility of defect size measurements acquired from 2D images needs to be validated. This study fulfills the requirement to combine all of these necessities into a model, and performs the validation of used methods via 3D rendering. It contains automatic detection solution which produces a faster, cheaper and most importantly a more consistent evaluation of defects.

In order to process the defect size estimation, the image type differentiation was needed (pore nest and overview image). After assigning an image to the right group according to its black/white pixel ratio (with 90% accuracy), noise reduction and thresholding is applied. For that purpose, the best performance was achieved with non-local denoising and Otsu thresholding.

Afterwards, the contours are found with *findContours* function of opencv library. The detected outer contours of the image and the metal are eliminated with a contour elimination process. The contour properties such as area, circumscribed diameter, perimeter and centroid are saved to use in clustering pores into pore groups.

The implemented defect size estimation methods contain different algorithms. It has two main aspects: clustering and size estimation. For the size of a pore network, “bounding rectangle” and “pore area model” were tested. The pore area model yields substantially

lower values than manual estimates, which makes it ineligible. For clustering the pores into pore networks, “K-means”, “K-means with PCA analysis”, “distance criteria method in bounding rectangle algorithm” are compared.

K-means approach has the drawback that the amount of clusters determines the performance and the optimal value varies in each image. K-means with PCA analysis takes also the shape of the pores into account, which has the advantage in following the rules of manual estimation but then the result is higher than expected values, since the distribution of the pores are dispersed. For finding the pore groups as in manual estimation, the best algorithm was bounding rectangle with the threshold “d+D”. It calculates the defect size of an image in around 55 seconds with around 10% error.

There is also another approach “Hough transformation”, which is applied directly to the contours and does not require the clustering step. The size of the line determines the size of the defect. Hough transformation is the fastest approach with around 6 seconds per image and is the best alternative method to bounding rectangle algorithm. However, it fails in finding reasonable results with overview images.

The structural analysis with image processing methods revealed that the average defect size among the image dataset is less than $1mm$. The comparison results between porosity percentage and defect size point out no direct correlation between them. Furthermore, the microstructural parameters such as perimeter and max. Feret diameter do not have a dependency on the porosity percentage. For a single pore size representation, max. Feret diameter is a better metric in pore size estimation.

The comparison between bounding rectangle results and validation via 3D renderings proved that the algorithm with “d+D” distance threshold value performs the most valid and plausible defect size estimation. The threshold had a recall 80% and a precision 85%.

This study achieved to compare the used methods with the analysis of 3D rendering images. The calculated ESD in 2D is not realistic and should be larger than estimated. Porosity percentage also differs in 2D and 3D. Hence, the results from 2D for porosity percentage should be adjusted to a lower value, if one wants to achieve 3D porosity predictions.

The analysis and evaluation on the life duration of components according to the defect sizes remains as future work. Fatigue testing and a study of the fracture surfaces of specimens are the next steps in the project. Furthermore, the evaluation can be generalized with other metal samples.

Since various image processing algorithms are already implemented in the automated tool, they can be also improved further. For reducing the computation time, clustering algorithms can be evolved more efficiently. For fully automated approach, the other types of component images can be analyzed and integrated into the tool.

List of Figures

1.1	Distribution of titanium and nickel alloys in an aero-engine. The titanium finds its use in the cooler parts of the engine (fan and compressor), since it is stable to approximately 880°C. The nickel-based superalloys run in the hotter parts of the engine (combustor chamber and turbine), which can reach up to 1000°C [ICC ⁺ 19].	2
2.1	The image on the left-hand side illustrates the raising gas concentration w_g^l in the mushy region based on gas microsegregation and the diminution of the solubility limit w_g^{l*} . The image on the right-hand side depicts the liquid pressure drop in the mushy region based on solidification shrinkage. The parameter ρ_l is the specific mass of the liquid, T is the temperature and v_l is the velocity of the grain. The figures are taken from [RMB16].	5
2.2	The image on the left-hand side demonstrates the internal pores, whereas the image on the right-hand side shows the surface pores. The blue marks represent porosity. The figure is taken from [WYZ16].	6
2.3	The component shown is simultaneously loaded by a force F and a torsional moment M_T . The fatigue crack grows perpendicular to the principal normal stress σ_1 [RS09].	6
2.4	Lengths and widths of a pore are analyzed. Maximum Feret is also to be found on the figure [Pic17].	7
2.5	This figure shows two pores represented by the two circles i, j and the distance a between them.	7
2.6	The left and right circles represent pores: A_B is the area of the bigger pore and A_S of a smaller pore. The dashed circle in the middle is the “imagined” pore which has the area A_{Im} . The method considers the half of each area.	9
2.7	(a) Illustration of the <i>non-local mean denoising</i> , (b) the original and (c) denoised light micrographs containing subsurface cracks. The figure is taken from [ZYSL18].	11
2.8	(a) Illustration of a gray scale image, (b) binary image obtained with Otsu thresholding algorithm. The figure is taken from [RIMB19].	12
2.9	An example of an edge detection algorithm operation. The figure is taken from [Kho12].	12

2.10	The left image shows the original image and on the right image the outer and inner contours are marked with blue and red colors, respectively. Figure is from [Tom20].	13
2.11	Implementation of dilation on the input image on the left-hand side. Figure is from [AAK00].	15
2.12	The red line marks the single pore in a specimen. The dotted yellow line shows the bounding right triangle. Figure is from [YMM ⁺ 17].	15
2.13	A line (red) can be represented in its polar form with another line which passes through the origin (blue). Its length from origin to the main line r and the angle of this line to the x-axis θ describes the line in polar form.	16
2.14	This figure highlights the relationship between Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL). Some example approaches of ML and DL are also given in the picture. Figure is from [O'R21].	17
3.1	This figure shows overall pore analysis and defect size estimation workflow. The workflow is divided into three main parts: (1) Structural analysis of contours with the preprocessing and contour detection steps, (2) Defect size estimation with determining the biggest cluster and determining the defect size (3) Validation with obtaining the 3D model and comparing the results with 2D analysis.	21
3.2	The image on the left is an example of an overview image, whereas the image on the right shows a pore nest image.	23
3.3	The images (c), (e), (g) and (i) show a pore nest image (a) denoised by different algorithms. The images (d), (f), (h), (j) show the same process on an overview image (b) accordingly. For the non-local denoising, h value is chosen as 30, template window size is 7 and search window size is 21. The used Gaussian kernel size here is (5,5), aperture linear size for median blur is 5. For bilateral filter is the sigma color 9, sigma space is 75 and border mode is 75. These input parameters are used in order to render the sample images with opencv library.	25
3.4	First row of images show an pore nest image both global and adaptive thresholding applied. Second row shows the same process on an overview image accordingly. For the global threshold 127 is chosen. For adaptive threshold, size of a pixel neighborhood that is used to calculate a threshold value for the pixel is 11.	26
3.5	The image on the left shows a pore nest image with Otsu thresholding applied. Right image shows the same process on an overview image accordingly.	27
3.6	Different edge detection algorithms are applied to a sample image. For Canny edge detection, the first threshold for the hysteresis procedure is 100 and the second threshold for the hysteresis procedure is 200. For Sobel filters, kernel size is 5.	28

3.7	The image on the left shows a pore nest image with <i>findContours</i> function applied and the detected contours are drawn on the image. Right image shows the same process on an overview image accordingly.	29
3.8	Closing morphology (dilation followed by erosion) applied to the image with the kernel size (8,8).	29
3.9	The pore nest image shows a pore nest with marked lines in inter-pore area. One can recognize the pores which are perpendicular to the main defect orientation.	30
3.10	Convex hull of a bounded planar set.	31
3.11	The image (a) is created with probabilistic Hough transformation after applying Canny edge detection algorithm. The image (b) displays detected lines on denoised images using the probabilistic Hough transformation and the <i>findContours</i> algorithm.	32
3.12	Red and blue lines represent the Silhouette score of two different images with various number of clusters. In general Silhouette score differs between 0.4 and 0.5. The best score is achieved with 4 clusters for one image and with 5 clusters for the other, whereas it slightly differs from the score of 4 clusters.	33
3.13	Both images show the correlation between <i>SSE</i> value and the number of clusters. The number of clusters is ranging from 1 to 10.	34
3.14	The images (a), (b) show the centers of contours assigned into two different clusters represented in red and purple colors. The images (c), (d) and (e), (f) show the contour centers assigned into 4 and 5 clusters, respectively. . .	36
3.15	Each PC is shown with the variance percentage with bar plot. The sum value of variances results in 100 %. To get the variances, 50 images are checked, 3120 contours are remained after detection and elimination process. Therefore the high dimensional matrix is 3120 x 20.	38
3.16	The heatmap shows for each PC, the correlations between the original variables. The right-hand bar shows the values that colors refer to. A positive correlation is shown by a positive value, and negative correlation by a negative value.	39
3.17	Red and blue lines represent the Silhouette score of two different images with $n = 2, 3, 4, 5$ number of clusters. In general, the Silhouette score differs between 0.7 and 0.9. The best score is achieved with 5 clusters for both images.	40
3.18	This image is obtained by the image tool Fiji. The same colors depict the same pores. To create this image, a maximum image stack of 2GB can be used in Fiji and this process takes approx. 20 hours.	42
4.1	The image (a) shows a pore nest image, which was assigned to the overview class. The overview image (b) was detected as pore nest image.	45

4.2	The image shows for 75 pore nest and 65 overview images the manually estimated defect sizes. Red points represent overview images and black circles represent pore nest images.	46
4.3	This plot shows max. Feret diameters of the detected contours in 140 component images. Black circles represent the mean value of max. Feret diameters in each image, whereas the red points represent the maximum value. They are depicted in relation to the porosity percentage.	47
4.4	This plot shows perimeters of the detected contours in 140 component images. Black circles represent the mean value of the perimeters in each image, whereas the red points represent the maximum perimeter value. They are depicted in relation to the porosity percentage.	47
4.5	First image on the left-hand side shows the bounding rectangle around the detected pores with “d” threshold. Second image shows with “d+D” and third image shows with “2D” accordingly.	48
4.6	This plot shows the defect sizes estimated with different thresholds in comparison with manually estimated defect sizes. The green line is perfect match line where the light and dark brown colors depict 20% and 40% error lines, accordingly.	49
4.7	This plot shows defect sizes of 140 component images. Red points represent the manual estimation results, whereas black circles points represent the defect sizes calculated with pore area model. They are depicted in relation to the porosity percentage of the image.	50
4.8	The figure (a) shows an overview image with the longest detected line of 4625 μm by the Hough transformation algorithm. The image(b) shows a pore nest image with the longest detected line of 1038 μm . The red lines represent the detected lines with minimum line length of 10 pixels and maximum line gap of 250 pixels.	51
4.9	This plot shows defect sizes of 140 component images. Black circles represent the manual estimation results, whereas red points represent the defect sizes calculated with Hough transformation. They are depicted in relation to the porosity percentage of the image.	51
4.10	The images on the left show the manually checked images. On the image (a), the biggest cluster has the size 1056 μm and the image (c) has the value of 783 μm . The images (b) and (c) show the K-means algorithm applied to the centroids of the detected contours. The red points represent the four cluster centers and other colors represent different clusters, respectively. . .	52
4.11	PCA analysis with K-means algorithm are applied to both images. The contours marked with purple lines are the biggest cluster resulted from the algorithm.	53
4.12	This representation is obtained with 20x8x2 subsampling. The illustrated pores have a minimum volume of 10.000 μm^3	53
4.13	This representation is obtained with 20x8x2 subsampling. The illustrated pores have the minimum volume of 100 μm^3	54

4.14	This representation is obtained with $20 \times 5 \times 1$ subsampling. The illustrated pores have the minimum volume of $10.000 \mu\text{m}^3$.	54
4.15	The figure presents one of the layer in 2D (a) with 3D model of the sample (b). The microstructures are depicted as 3D surface mesh of triangles with Paraview software.	55
4.16	This plot shows the defect sizes estimated with different thresholds in comparison with manually estimated defect sizes. The green line is perfect match line where the light and dark brown colors depict 20% and 40% error lines, accordingly.	57
4.17	An example of a single pore (a) in XY axis, (b) in ZY axis and (c) shows one slice of this pore where all axis can be seen. The pore is selected from Sample 3.	59
4.18	This plot shows an example image with “d” threshold estimated bounding rectangles.	60
4.19	The images (a)-(e) show different layers of Sample 3. The red, green and blue marked areas highlight the same pore groups, which appear differently on different layers.	61

List of Tables

4.1	Microstructural analysis 2D vs. 3D	55
4.2	Performance of different threshold settings	56
4.3	Microstructural analysis 2D vs. 3D	59

Chapter 6

Abbreviations

AI	Artificial Intelligence
approx.	approximately
CNN	Convolutional Neural Network
ct	computer tomography
CSV	Comma-Separated Values
DL	Deep Learning
ESD	Equivalent Spherical Diameter
FN	False Negative
FP	False Positive
GB	Gigabyte
HSV	Hue Saturation Value

IMS	Image Management System
IN-718	Inconel 718
IN-713	Inconel 713
LP	Large porosity
LSTM	Long Short Term Memory
ML	Machine Learning
NaN	Not a Number
PC	Principal Component
PCA	Principal Component Analysis
RF	Random Forest
RGB	Red Green Blue
SP	Small porosity
SSE	Sum of Squared Error
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VAR	Variance

XML Extensible Markup Language

3D Three dimensional

2D Two dimensional

Bibliography

- [AAK00] Nasser Kamiss Al-Ani and Tomasz Kacprzak. Time-varying cellular neural network based morphological image processing. In *Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA 2000)(Cat. No. 00TH8509)*, pages 87–92. IEEE, 2000.
- [BA96] J Martin Bland and Douglas G Altman. Statistics notes: measurement error. *Bmj*, 312(7047):1654, 1996.
- [BP12] Himani Bhavsar and Mahesh H Panchal. A review on support vector machine for data classification. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(10):185–189, 2012.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [BR4] W. J. BRÄUNLING. *Flugtriebwerke*. Springer VDI, 2004.
- [CM05] Bartomeu Coll and Jean-Michel Morel. A non-local algorithm for image denoising. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:60–65, 07 2005.
- [CMY⁺17] Andrew Campbell, P Murray, Evgenia Yakushina, S Marshall, and William Ion. Automated microstructural analysis of titanium alloys using digital image processing. *IOP Conference Series: Materials Science and Engineering*, 179:012011, 02 2017.
- [Cor20] Corrosionpedia. Cyclic loading. <https://www.corrosionpedia.com/definition/6333/cyclic-loading>, 2020. Last accessed: May 22, 2021.
- [Doc17] Opencv Documentation. class cv::moments. https://vovkos.github.io/doxyrest-showcase/opencv/sphinx_rtd_theme/class_cv_Moments.html, Copyright 1999-2017. Last accessed: August 2021.
- [DYL⁺17] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *annual conference on medical image understanding and analysis*, pages 506–517. Springer, 2017.

- [Eli18] Frederik Elischberger. Image segmentation for the microstructural analysis of engine materials. Masterthesis, Westfälische Wilhelms-Universität, Münster, 2018.
- [Für05] W Fürbeth. Buchbesprechung: Metallografie. von h. schumann und h. oettel. *Chemie Ingenieur Technik*, 77(4):443–444, 2005.
- [GLK97] L. Garimella, P. Liaw, and D. Klarstrom. Fatigue behavior in nickel-based superalloys: A literature review. *JOM*, 49:67–71, 1997.
- [GSTM92] AK Gupta, BK Saxena, SN Tiwari, and SL Malhotra. Pore formation in cast metals and alloys. *Journal of materials science*, 27(4):853–862, 1992.
- [GVG⁺17] Satya Ganti, Michael Velez, Brian Geier, Brian Hayes, Bryan Turner, and Elizabeth Jenkins. A comparison of porosity analysis using 2d stereology estimates and 3d serial sectioning for additively manufactured ti 6al 2sn 4zr 2mo alloy. *Practical Metallography*, 54:77, 02 2017.
- [GYLL04] YX Gao, J. YI, Peter Lee, and T. LINDLEY. The effect of porosity on the fatigue life of cast aluminum-silicon alloys. *Fatigue Fracture of Engineering Materials & Structures*, 27:559–570, 06 2004.
- [HBPL16] Korbinian Heim, F Bernier, R Pelletier, and L-P Lefebvre. High resolution pore size analysis in metallic powders by x-ray tomography. *Case Studies in Nondestructive Testing and Evaluation*, 6:45–52, 2016.
- [HPJ⁺21] Rokgi Hong, Jinseok Park, Seongju Jang, Hyungjin Shin, Hakkwan Kim, and Inhong Song. Development of a parcel-level land boundary extraction algorithm for aerial imagery of regularly arranged agricultural areas. *Remote Sensing*, 13(6):1167, 2021.
- [HYS16] Nur Farhana Hordri, Siti Sophiayati Yuhaniz, and Siti Mariyam Shamsuddin. Deep learning and its applications: a review. In *Conference on Postgraduate Annual Research on Informatics Seminar*, 2016.
- [ICC⁺19] E.E. Ibrahim, M. Checkley, Xun Chen, Martin Sharp, W. Liang, S. Yuan, and Andre Batako. Process performance of low frequency vibratory grinding of inconel 718. *Procedia Manufacturing*, 30:530–535, 01 2019.
- [IS05] Bernhard Ilchner and Robert F Singer. *Werkstoffwissenschaften und Fertigungstechnik*. Springer, 2005.
- [JC16] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.
- [Kho12] M Yu Khomyakov. Comparative evaluation of linear edge detection methods. *Pattern Recognition and Image Analysis*, 22(2):291–302, 2012.

- [Kra13] Carolin Krammer. Korrelation der flächenporosität mit der gröÙe des risskeims in der nickelbasis-superlegierung mar-m247. Masterthesis, Technical University of Munich (TUM), 2013.
- [LAMS16] K Lalitha, R Amrutha, S Michahial, and M Shivakumar. Implementation of watershed segmentation. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(12), 2016.
- [LEH02] T Link, A Epishin, and A Haibel. Investigation of microporosity in single-crystal nickel-base superalloys by different experimental techniques. *Proceedings of EUROMAT 2005*, 2002.
- [LKA17] Jake Lever, Martin Krzywinski, and Naomi Altman. Points of significance: Principal component analysis. *Nature methods*, 14(7):641–643, 2017.
- [Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [LWS⁺19] Yang Liu, Lei Wang, Xiu Song, Taosha Liang, and Guo Hua. Effects of treatment on microstructure and deformation behavior of dissimilar welded joint between single crystal and polycrystalline superalloy. In *International Conference on Theoretical, Applied and Experimental Mechanics*, pages 48–53. Springer, 2019.
- [LWW⁺20] Y. Lu, M. Wang, Z. Wu, I. P. Jones, M. Wickins, N. R. Green, and H. C. Basoalto. Three-dimensional analysis of dendrites via automated serial sectioning using a robo-met.3d. *MRS Communications*, 10(3):461–466, 2020.
- [MB99] Y Murakami and Stefano Beretta. Small defects and inhomogeneities in fatigue strength: experiments, models and statistical implications. *Extremes*, 2(2):123–147, 1999.
- [MBH⁺18] Boyuan Ma, Xiaojuan Ban, Haiyou Huang, Yulian Chen, Wanbo Liu, and Yonghong Zhi. Deep learning-based image segmentation for al-la alloy microscopic images. *Symmetry*, 10(4):107, 2018.
- [ME94] Y. Murakami and M. Endo. Effect of defects, inclusions and inhomogeneities on fatigue strength. *Int. J. Fatigue*, 16:163–181, 1994.
- [MMA20] Dastan Maulud and Adnan Mohsin Abdulazeez. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1:140–147, 12 2020.
- [MS14] AM Morad and YM Shash. Nickel base superalloys used for aero engine turbine blades. In *The International Conference on Applied Mechanics and Mechanical Engineering*, volume 16, pages 1–22. Military Technical College, 2014.

- [Mur85] Y Murakami. Analysis of stress intensity factors of modes i, ii and iii for inclined surface cracks of arbitrary shape. *Engineering Fracture Mechanics*, 22(1):101–114, 1985.
- [NKF12] Gianni Nicoletto, Radomila Konečná, and Stanislava Fintova. Characterization of microshrinkage casting defects of al–si alloys by x-ray computed tomography and metallography. *International journal of fatigue*, 41:39–46, 2012.
- [NPaST19] Rena Nainggolan, Resianta Perangin-angin, Emma Simarmata, and Astuti Feriani Tarigan. Improved the performance of the k-means cluster using the sum of squared error (sse) optimized by using the elbow method. In *Journal of Physics: Conference Series*, volume 1361, page 012015. IOP Publishing, 2019.
- [Ope20a] Opencv. Contours.cpp. <https://github.com/opencv/opencv/blob/master/modules/imgproc/src/contours.cpp>, 05 2020. Last accessed: August 2021.
- [Ope20b] Opencv. Image thresholding. https://docs.opencv.org/4.5.1/d7/d4d/tutorial_py_thresholding.html, 12 2020. Last accessed: August 2021.
- [Ope21a] Opencv. 2d convolution (image filtering). https://docs.opencv.org/4.5.2/d4/d13/tutorial_py_filtering.html, 04 2021. Last accessed: August 2021.
- [Ope21b] Opencv. Hough line transform. https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html, 04 2021. Last accessed: August 2021.
- [Ope21c] Opencv. Morphological transformations. https://docs.opencv.org/4.5.2/d9/d61/tutorial_py_morphological_ops.html, 04 2021. Last accessed: August 2021.
- [O’R21] O’Reilly. A brief overview of deep learning. <https://www.oreilly.com/library/view/python-natural-language/9781787121423/3a3082d5-ecc5-4b2a-bee5-c9636d63710d.xhtml>, 2021. Last accessed: July 2021.
- [OSC⁺20] L.M. Bortoluci Ormastroni, L. Mataveli Suave, A. Cervellon, P. Villechaise, and J. Cormier. Lcf, hef and vhef life sensitivity to solution heat treatment of a thirdgeneration ni-based single crystal superalloy. *Int. J. Fatigue*, 130, 2020.
- [Pei20] Vidyaratne L. Rahman M.M. et al. Pei, L. Context aware deep learning for brain tumor segmentation, subtype classification, and survival prediction using radiology images. *Sci Rep*, 10, 2020.
- [PG07] W Pabst and E Gregorova. Characterization of particles and particle systems. *ICT Prague*, 122:122, 2007.
- [PGC⁺19] Emeric Plancher, Pauline Gravier, Edouard Chauvet, Jean-Jacques Blandin, Elodie Boller, Guilhem Martin, Luc Salvo, and Pierre Lhuissier. Tracking pores during solidification of a ni-based superalloy using 4d synchrotron microtomography. *Acta Materialia*, 181:1–9, 2019.

- [Pic17] Vincent Picandet. *Particle Size Distribution*, pages 91–110. Springer Netherlands, Dordrecht, 2017.
- [PŚ13] M Perzyk and SJ Świłło. Surface casting defects inspection using vision system and neural network techniques. *Archives of foundry engineering*, 2013.
- [Pyt18] Python. Python 3.7.0. <https://www.python.org/downloads/release/python-370/>, 06 2018. Last accessed: October 2021.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [RHH19] SM Mahbubur Rahman, Tamanna Howlader, and Dimitrios Hatzinakos. Image moments and moment invariants. In *Orthogonal Image Moments for Human-Centric Visual Pattern Recognition*, pages 19–48. Springer, 2019.
- [RIMB19] Farzaneh Rezaei, Hossein Izadi, Hossein Memarian, and Majid Baniassadi. The effectiveness of different thresholding techniques in segmenting micro ct images of porous carbonates to estimate porosity. *Journal of Petroleum Science and Engineering*, 177:518–527, 2019.
- [RMB16] Michael Riedler, Sebastian Michelic, and Christian Bernhard. Formation of shrinkage porosity during solidification of steel: Numerical simulation and experimental validation. In *IOP Conference Series: Materials Science and Engineering*, volume 143, page 012035. IOP Publishing, 2016.
- [RNF18] J Rowenhorst, Lily Nguyen, and Richard W Fonda. 3d analysis of large volumes through automated serial sectioning david. *Microscopy and Microanalysis*, 24(S1):552–553, 2018.
- [RNR16] R Pradeep Kumar Reddy, Chiluka Nagaraju, and I Rajasekhar Reddy. Canny scale edge detection. *International Journal of Engineering Trends and Technology (IJETT)*, 2016.
- [Ros13] Stanisław Roskosz. Evaluation of porosity of precision castings made of high-temperature creep resisting nickel superalloys. *Practical Metallography*, 50:527–547, 08 2013.
- [Rou87] Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.
- [RS09] Hans Albert Richard and Manuela Sander. *Ermüdungsrisse*. Springer, 2009.
- [S⁺85] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

- [Sen20] Carter J.L.W. Senanayake, N.M. Automated microstructural analysis of titanium alloys using digital image processing. *Integrating Materials and Manufacturing Innovation*, 9:446–458, 02 2020.
- [SMP03] Jonathan E Spowart, Herbert E Mullens, and Brian T Puchala. Collecting and analyzing microstructures in three dimensions: a fully automated approach. *Jom*, 55(10):35–37, 2003.
- [SRRL⁺20] Ankita Singh, Arash Rabbani, Klaus Regenauer-Lieb, Ryan Armstrong, and Peyman Mostaghimi. Computer vision and unsupervised machine learning for pore-scale structural analysis of fractured porous media. *Advances in Water Resources*, 147, 11 2020.
- [SSS⁺19] T.M. Smith, N.M. Senanayake, C.K. Sudbrack, P. Bonacuse, R.B. Rogers, P. Chao, and J. Carter. Characterization of nanoscale precipitates in superalloy 718 using high resolution sem imaging. *Materials Characterization*, 148:178–187, 2019.
- [SYS00] LK Shark, C Yu, and JP Smith. Automatic estimation of ultrasonic attenuation for porosity evaluation in composite materials. *ROMA. 15th WCNDT*, 2000.
- [TAL16] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.
- [Tom20] Tomole. Detect overlapping noisy circles in image. <https://i.stack.imgur.com/PVpWV.jpg>, 05 2020. Last accessed: July 2021.
- [Tsa18] Nika Tsankashvili. Comparing edge detection methods. <https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e>, 1 2018. Last accessed: June 2021.
- [Uch13] Seiichi Uchida. Image processing and recognition for biological images. *Development, growth & differentiation*, 55(4):523–549, 2013.
- [Vyn20] M. Vynnycky. On the formation of centreline shrinkage porosity in the continuous casting of steel. *Journal of Mathematics in Industry*, 10, 12 2020.
- [WKH⁺18] Si Wen, Tahsin Kurc, Le Hou, Joel Saltz, Rajarsi Gupta, Rebecca Batiste, Tianhao Zhao, Vu Nguyen, Dimitris Samaras, and Wei Zhu. Comparison of different classifiers with active learning to support quality control in nucleus segmentation in pathology images. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2017:227–236, 05 2018.
- [WLY⁺21] Jiang Wen, Piao Li, Wei-Xing Yao, Shao-Shi Rui, Hui-Ji Shi, and Jie Huang. The effect of porosity size on the high cycle fatigue life of nickel-based single crystal superalloy at 980°C. *International Journal of Fatigue*, 147:106191, 02 2021.

- [WVD89] M. WEVER, I. VERPOEST, and P. DE MEESTER. Identification of fatigue failure modes in carbon fibre reinforced composites. In J. Boogaard and G.M. van Dijk, editors, *Non-Destructive Testing*, pages 459–465. Elsevier, Oxford, 1989.
- [WYZ16] Yong Wang, Wenzheng Yue, and Mo Zhang. Numerical research on the anisotropic transport of thermal neutron in heterogeneous porous media with micron x-ray computed tomography. *Scientific reports*, 6(1):1–9, 2016.
- [YMM⁺17] Yoichi Yamashita, Takao Murakami, Rei Mihara, Masami Okada, and Yukitaka Murakami. Defect analysis and fatigue design basis for ni-based superalloy 718 manufactured by additive manufacturing. *Procedia Structural Integrity*, 7:11–18, 2017.
- [ZES⁺15] Sahar Zafari, Tuomas Eerola, Jouni Sampo, Heikki Kalviainen, and Heikki Haario. Segmentation of overlapping elliptical objects in silhouette images. *IEEE Transactions on Image Processing*, 24:1–1, 10 2015.
- [ZYSL18] Yong Jie Zhao, Yun Hui Yan, Ke Chen Song, and Hao Nan Li. Intelligent assessment of subsurface cracks in optical glass generated in mechanical grinding process. *Advances in Engineering Software*, 115:17–25, 2018.