CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Data Analysis for Defect Monitoring in Additive Manufacturing

Applying Machine Learning to Predict Porosity in L-PBF

Master's thesis in Computer science and engineering

Erik Sievers

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Data Analysis for Defect Monitoring in Additive Manufacturing

Applying Machine Learning to Predict Porosity in L-PBF

Erik Sievers

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Data Analysis for Defect Monitoring in Additive Manufacturing
Applying Machine Learning to Predict Porosity in L-PBF
Erik Sievers

Supervisor: Marina Papatriantafilou, Computer Science and Engineering
Examiner: Vincenzo Gulisano, Computer Science and Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2023

Data Analysis for Defect Monitoring in Additive Manufacturing
Applying Machine Learning to Predict Porosity in L-PBF
Erik Sievers
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Laser powder bed fusion (L-PBF) is an additive manufacturing technique that sees more and more use in industrial settings, but is held back by a lack of cost-effective quality validation of created products. One core attribute of high-quality additive manufactured products is a low porosity, i.e. a high ratio of solid to empty volume inside the object. This thesis provides an overview of the state of the art for in-situ monitoring of L-PBF manufacturing and investigates the use of outlier detection methods as a way of encoding optical tomography data from an L-PBF process. This is done using a commercial L-PBF machine with its accompanying in-situ monitoring camera. The results show that outlier detection methods can be used to detect porosity in created objects (0.94 - 0.99 ROC-AUC, receiver operating characteristics' area under curve) and that it can generalize between similar object geometries. The thesis also provides a discussion of the limitations of the current research and suggests future work both building upon the methods introduced in the thesis and in the field of in-situ monitoring of L-PBF.

Keywords: Machine Learning, Outlier Detection, Additive Manufacturing, Powder Bed Fusion, Optical Tomography, Porosity

Acknowledgements

I would like to thank my supervisor, Marina Papatriantafilou for her guidance, support and invaluable feedback throughout the thesis. The thesis would not have happened without her support, for which I'm deeply grateful. Furthermore, I would like to thank Zhouer Chen for his input and suggestions and providing a material science perspective and Negar Panahi for allowing me to use images from her thesis. Finally, I'd like to thank my family, friends and co-workers for their support and encouragement throughout the writing process.

Erik Sievers, Gothenburg, March 2023

Contents

List of Figures

List of Tables

Glossary

- **Additive Manufacturing (AM)** - Manufacturing method where the object is created by adding instead of removing material.
- **Artificial Neural Network (ANN)** - A type of supervised machine learning model.
- **Build Plate** - The plate in an L-PBF machine upon which the powder is spread to form the *powder bed*. The build plate is lowered after each layer to keep the top of the powder bed at the same height.
- **Clustering** - A type of unsupervised machine learning method.
- **Cross validation** - A method used in machine learning to determine suitable parameters for a model.
- **Co-axical** - Set up where the camera is aligned with the laser beam, so that the camera follows the laser spot. See also *off-axial*.
- **Gas pore** - A pore that occurs as a result of gas being trapped in the powder used. It is usually small and fairly spherical.
- **Keyhole Pore** - A pore that occurs in the **melt pool**, often as a consequence of the *volumetric energy density (VED)* being too high. It is usually fairly large and fairly spherical.
- **k-nearest neighbour classifier** - A machine learning model that classifies instances according to the class of the k-nearest neighbours, with k being a parameter (typically set to 5).
- **Lack-of-Fusion (LOF) Pore** - A pore that occurs as a result of the powder not melting completely. It is usually fairly large and has an irregular shape.
- **Laser Powder Bed Fusion (L-PBF)** - An additive manufacturing method using a laser to melt layers of metal powder together to form an object.
- **Melt Pool** - The volume of powder melted by the laser.
- **Off-axical** - Set up where the camera angle is static, observing the whole *powder bed* at the same time. See also *co-axical*

- **Optical Tomography (OT)** - A method that creates a three-dimensional model of an object by combining multiple images of an object. Created using an *off-axical* camera set up.
- **Porosity** - The empty space inside of an object.
- **Powder Bed** - The area in an L-PBF machine where the powder is spread and the object is built.
- **Volumetric Energy Density (VED)** - A measurement of the amount of energy used per volume unit.

1
Introduction

1.1 Additive Manufacturing

Laser Powder Bed Fusion (L-PBF) is a manufacturing method where an object is created by melting layers of a fine metal powder on top of each other, one after another, according to a specification (often a CAD file). It is an *additive manufacturing* method, i.e. a method where instead of creating an object by removing parts from a larger chunk, it is created by adding material. L-PBF initially mainly saw use in prototyping, but increasingly sees use in production settings as well.
Although L-PBF offers a number of benefits compared to traditional metal manufacturing methods such as less waste, faster prototyping and the possibility to create unique shapes, the technology is still held back by a lack of process repeatability and inconsistent quality of build objects due to defects arising during the manufacturing process [9].
One type of defect, *porosity* (the amount of empty space inside an object), can severely impact the durability and toughness of a component [3]. Although some degree of porosity is to be expected due to impurities in the material used, limiting the occurrence of pores is of high importance to ensure the reliability of objects created using L-PBF.
Traditionally, the way to address the issue has been in an *open-loop* fashion [14]. This means that an object would be manufactured, then checked for faults. If these faults are considered severe enough, the object is discarded and the parameters controlling the manufacturing process are tweaked if needed. This approach is time-consuming (since it adds a qualifying step in the manufacturing of a component) and if a critical issue occurs in an early layer, the object is still finished before it is evaluated.

1.2 In-situ monitoring

To overcome these issues and increase the viability of L-PBF at scale, there has been a surge in research of *in-situ* monitoring. Instead of waiting until the component is completed, in-situ monitoring aims to evaluate the component as it is being constructed. In addition to increasing production speed and reducing material waste, in-situ monitoring could potentially aid in taking corrective action when defects occur, reducing the amount of constructed objects that are discarded.
Any in-situ monitoring system consists of at least two parts: a sensing part gathering data about the manufacturing process and an analysis part looking at making sense

of the gathered data. Multiple types of sensors have been used, some of the most frequently used ones being optical sensors such as cameras or photodiode based systems [3]. Although different researchers [23][24][6][15] have used similar sensing setups when looking at the whole build area (called the *powder bed*) and concluded that there is the potential to use this type of data in-situ monitoring, there is limited research in algorithmically analysing the sensor data to actually quantify porosity. The research that exists has focused on using *probability maps* (a statistical tool) to predict porosity in small parts of an object, and without taking the surrounding area or layers into account [11].

1.3 Spatial Data Mining and Machine Learning

One way information could be extracted from the kind of images produced by in-situ monitoring cameras is through the use of spatial data mining, the process of extracting patterns from spatial data. It spans a range of methods, some commonly sorted under *supervised machine learning* (such as *artificial neural networks*) as well as *unsupervised machine learning* (*clustering* being one example). Clustering in particular has seen a wide range of uses, such as identifying power outages [7], identifying dangerous stretches of road [18] and identifying credit card fraud [1].

1.4 Study objectives

This thesis contributes to the existing research by assessing the use of spatial data mining methods to automatically classify AM objects based on their porosity using *powder bed* images. The aim of the thesis is to predict the porosity of components manufactured using laser powder bed fusion by looking at in-situ images. Mohr et al. [15] found that higher porosity results in uneven thermal conductivity (i.e. different parts of the constructed object staying warmer for longer than the surrounding) and that this difference in thermal conductivity can be detected using optical tomography. This thesis identifies and quantifies this phenomenon using spatial data mining methods, compare their precision to each other as well as existing in-situ monitoring methods. In doing this, we face the following challenges:

- Producing training data is expensive, since this involves physically building objects. As a consequence, the training data is limited in terms of number of instances.
- Due to the limited size of the training data, the traditional approach for image classification (convolutional neural networks) is unlikely to be effective, and other means of encoding need to be used.
- The available research on how one layer in a build object affects the previous layers has been limited.

This thesis overcomes these challenges and contribute to existing research by looking at optical microscopy data, which is cheaper to obtain, that has been used in a different study. It also introduces a new way of encoding the data, based on *spatial outlier detection* methods, which also takes multiple layers into account.

2
Additive Manufacturing

This chapter explains *additive manufacturing* and *Laser Powder Bed Fusion (L-PBF)*. It serves as an introduction for readers with a computer science background to provide a bit of context for the rest of the thesis. It begins by providing an overview of additive manufacturing methods before moving on to explain the numerous parameters for L-PBF and how these parameters impact the process. Finally, the chapter explains some in-situ sensing methods relevant to the thesis.

2.1 Introduction to L-PBF

There are multiple different additive manufacturing methods that vary from each other in terms of material used, source of power and type of method used for constructing the object. Three forms of material (called *base*) are common: powder, solid and liquid. The types of methods can be further divided by the method used to fuse material together, such as melting, binding or lamination [4]. Figure 2.1 shows an overview of some AM methods according to this classification, with Laser Powder Bed Fusion (the method used in this thesis) highlighted in blue.

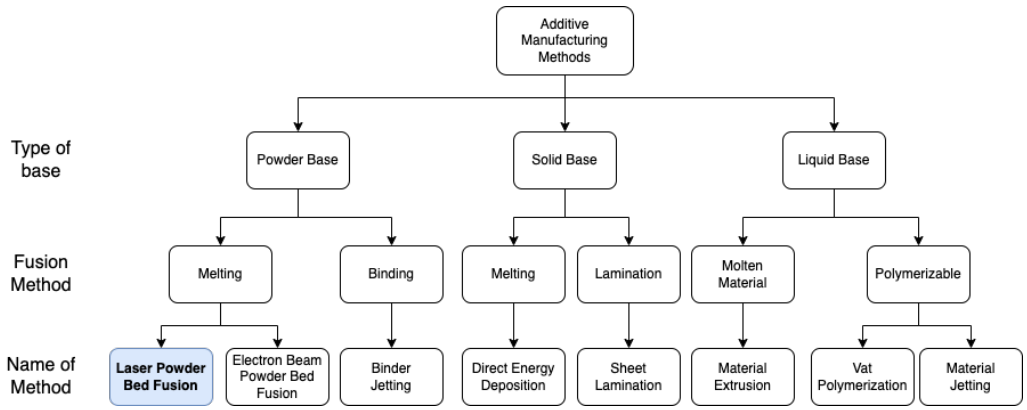


Figure 2.1: Laser Powder Bed Fusion (blue) in relation to other additive manufacturing methods. Figure adapted from Dharnidharka et al. [4]

Laser Powder Bed Fusion (L-PBF) is one of the more common methods for metal based additive manufacturing. L-PBF works in an iterative fashion, where an item is constructed in multiple layers. Each step begins with a *powder roller* spreading a thin layer of a fine metallic powder on top of the *powder bed*. After the powder has been spread, a powerful laser quickly scans across the layer, melting the powder,

to form a cross-section of the item under construction. After each layer, the object being constructed is lowered by the same thickness as the layer and the process is repeated, melting each layer to the previous layer. This process is repeated until the item under construction is complete. Figure 2.2 shows an overview of the L-PBF build chamber.

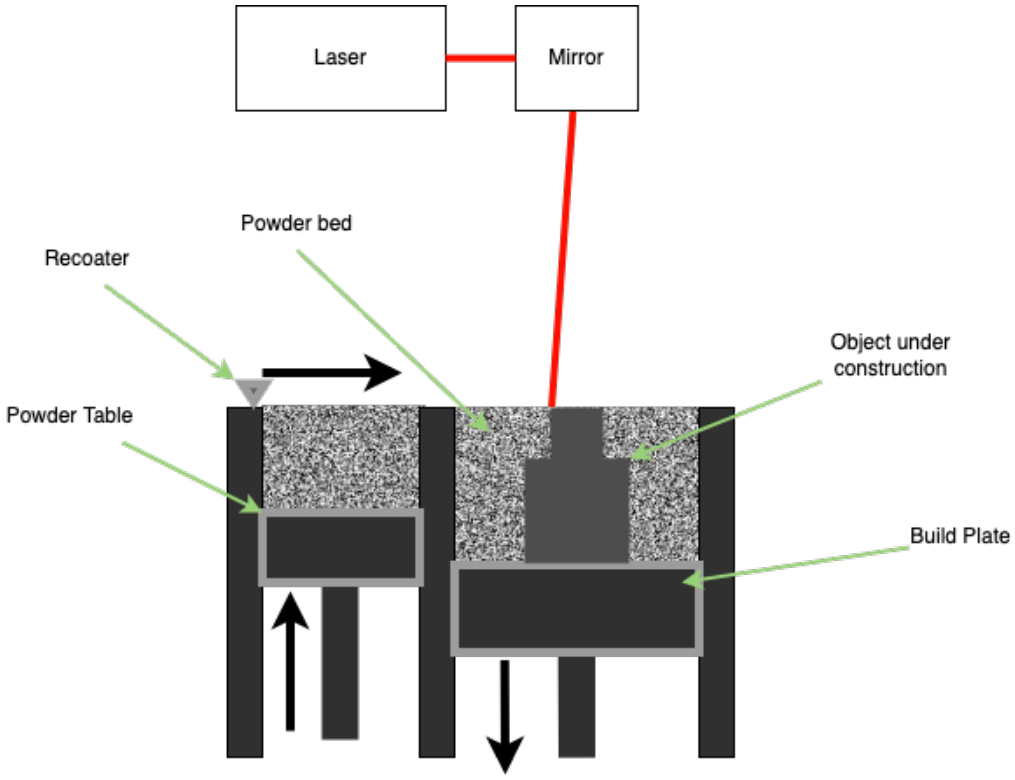


Figure 2.2: Overview of L-PBF process

2.2 Parameters in L-PBF

There are multiple parameters controlling a PBF process that need to be properly set in order to build objects properly. These parameters are commonly set according to established best practises or the manufacturer's recommendation in an open-loop fashion, meaning that the operator manually tunes parameters after the construction of an object depending on the quality of the last object [15]. Some of the more common parameters include:

- **Scan Speed**: The speed at which the laser tracks across the powder bed.
- **Powder depth**: The thickness of the powder in each layer
- **Hatch Distance**: The distance between the middle of two scan tracks of the laser
- **Laser Power**: The power of the laser beam
- **Volumetric Energy Density (VED)**: The energy used, measured in J/mm^3. It is composed of four other parameters: scan speed (the speed at which the laser scans across the object), laser power, powder layer thickness and hatch

distance (the distance between two parallel tracks in the laser scanning path). VED can be calculated as shown in equation 2.1, where P is the laser power in watt, S is the scan speed, H is the hatch distance and D is the powder layer thickness.

$$VED = \frac{P}{S * H * D} * 10^3 \tag{2.1}$$

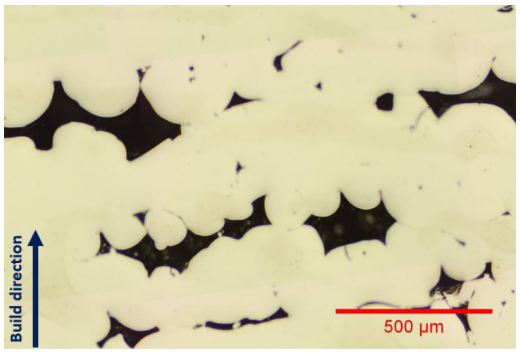
- **Beam Diameter**: The diameter of the laser/electron beam
- **Scan Strategy**: How the laser tracks across the object, for instance making a lot of parallel lines or alternating horizontal and vertical lines between layers
- **Printing angle**: Depending on where on the build plate an object is, the printing angle will be different. Directly under the power source the printing angle is 0 and the further from that location, the higher the angle will be
- **Air flow direction**: The direction of the air flow inside the build chamber
- **Air flow rate**: The rate of air flow through the build chamber
- **Powder recoating direction**: The direction the recoater spreads the powder between each layer
- **Material**: The build material. Different materials have different properties and require the other parameters to be tuned differently
- **Baseplate thickness**: The thickness of the baseplate (the plate the object is built upon). It needs to be sufficiently thick to allow proper thermal conductivity from the build object [9]

2.3 Defects in L-PBF

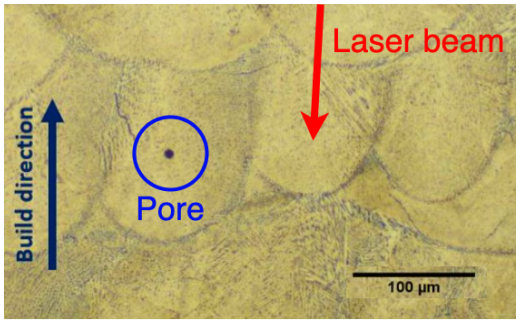
During an L-PBF process, various kinds of defects can occur, impacting the density or durability of the component being constructed. These defects can occur for a number of different reasons.

2.3.1 Pores and Porosity

Pores are voids appearing inside the component during the manufacturing process and can greatly impact the durability of a component. They are commonly split into three categories: *gas pores*, *lack-of-fusion (LOF) pores* and *keyhole pores*. Each of these pores are formed under different conditions, and they tend to have different physical characteristics. Still, these characteristics exist along a continuous spectrum, making it often hard or downright impossible to say what class a given pore belongs to. Figure 2.3 shows cross-sections from two objects with some examples of pores.



(a) Lack of fusion pores



(b) Melt pools and a single gas or keyhole pore. Red is the approximate direction of the laser

Figure 2.3: Two cross-sections of a built object. The black dots in the left image are pores with the typical, elongated and irregular shape of lack of fusion (LOF) pores. In the right image, the teeth or finger-like shapes are melt pools. The small, circular black dot is indicative of a gas or keyhole pore. Pictures taken from [19], with permission.

- *Gas pores* are the smallest kind of pores. They are created as a result of gas being trapped inside the powder. These pores are typically very small and spherical.
- *Keyhole pores* often occur as a result of the VED being set too high. They occur inside the *melt pool* (the small volume the laser heats to melt the metal). Like gas pores, keyhole pores tend to be fairly spherical, but they do however tend to be larger than gas pores. As a consequence, it is often hard to tell the difference between a keyhole pore and a gas pore.
- *Lack-of-fusion pores*, are named after how they are formed: they occur when there is a lack of fusion between the metal grains of the powder, which can occur if the VED is too low (resulting in the powder not melting completely), or if the hatch distance is too high (resulting in elongated pores between scan tracks). These pores typically have highly irregular shapes [22].

Since it is more important to fix larger pores and since gas pores are unavoidable, it is of interest to be able to classify pores as they occur. In a *closed-loop* setting (i.e. a setting where the process parameters are tweaked based on what happens during the build process), this may enable correcting pores as they occur since the laser parameters in future layers can heal previous layers [22][5].

2.3.2 Other Defects

In addition to porosity, there are a number of other defects that can occur during the additive manufacturing process. Below, we list some of the more common defects and their causes.

Balling Due to surface tension, the molten material in the melt pool may at times form small beads instead of a layer. The irregularity of these beads can result in lack of fusion pores (when occurring inside the object), increased surface roughness

(when occurring closer to the edge) as well as equipment damage in severe cases. Balling can occur as a consequence of the scan speed being too high [9].
Geometric defects The constructed part may have a different shape or size than the intended design. This can occur due to various reasons, including where on the build plate the object is located (since this impacts the printing angle) [2].
Surface Roughness The surface roughness of a component can often be improved by post-processing of the component (such as grinding). However, it can negatively impact the *fatigue performance*, i.e. how prone the object is to break due to stress [9]. A certain degree of surface roughness is to be expected due to the metal and the layer-by-layer method of construction (resulting in tilted surfaces getting a surface like a staircase), however it can become worse by pores occurring on the surface or balling defects happening close to the surface.

3

Data Mining and Machine Learning

This chapter explains the needed background knowledge within the field of data mining and machine learning. It starts by providing a brief introduction to the common terminology used in the field. It then goes on to explain different evaluation metrics, followed by a section on outlier detection with an emphasis on spatial outlier detection.

3.1 Data Mining and Machine Learning Overview

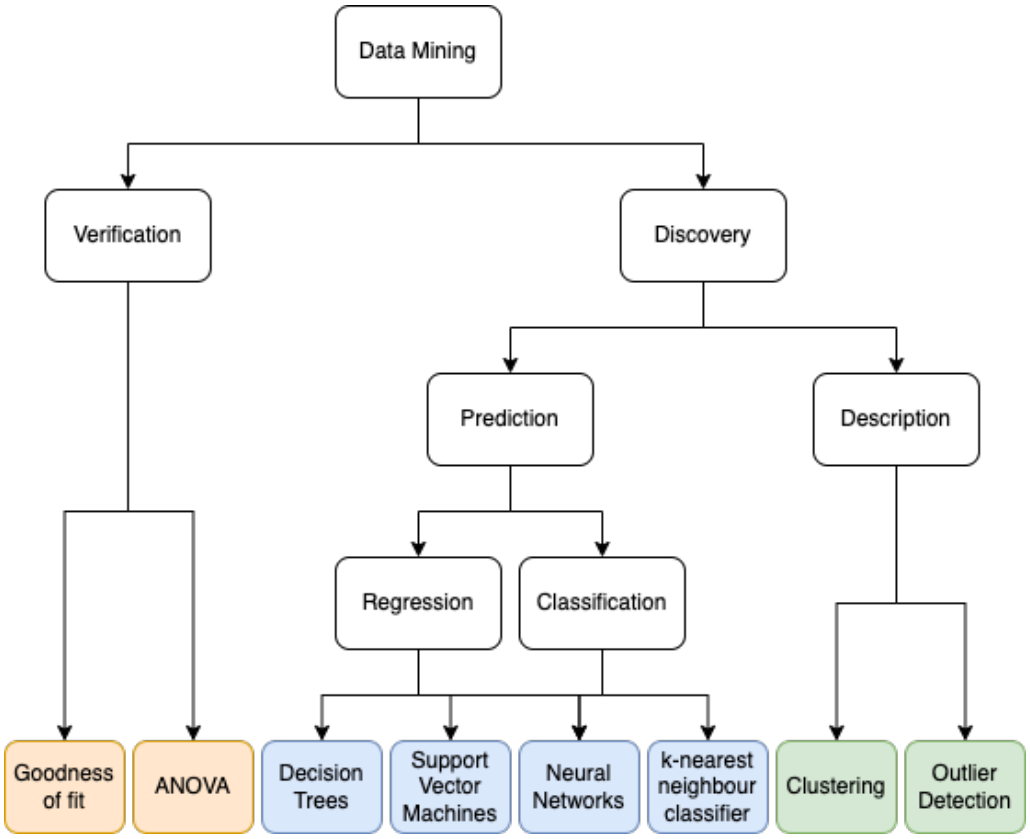


Figure 3.1: Overview of some data mining methods. Methods in orange are statistical methods, blue are supervised learning algorithms and green are unsupervised learning algorithms. Figure adapted from Maimon and Rokach [12]

.

Data Mining is a field of computer science concerned with making sense and finding connections in data. Maimon and Rokach [12] view the field as a hierarchy of categories and methods (figure 3.1), which can be split into two distinct categories: *verification* and *discovery*. Verification is about validating and testing existing hypothesis (typically created by experts in their field), and commonly uses traditional methods from statistics. Discovery on the other hand is about discovering a hypothesis, rather than testing one. Out of the two, discovery is the category that is more heavily associated with data mining. Discovery is in turn split in two categories of methods: *prediction* and *description* methods.
Prediction (referred to as *supervised learning* in machine learning terminology) algorithms attempt to find relationships in between input parameters (called *features*) and their corresponding output attribute (called the *label*). They do this by training a *model*. Although training a model often is time-consuming, once a model has been trained it can typically make predictions quickly [12].
Description algorithms aim at providing some insight into a data set when there is no clear "right" answer to a question. Some description methods (for instance

clustering) fall under the *unsupervised learning* field of machine learning. Others (for instance visualization) are not part of the machine learning field [12].

3.1.1 Regression and Classification Problems

Supervised learning models can further be split in two categories depending on the nature of the prediction they are making. A regression model is predicting a continuous value (for instance the price of a house given its location, size, number of rooms etc.), whereas a classification model is predicting what category a given instance belongs to (for instance whether a tumour is malign or whether someone will default on their bank loan) [12].
Although it may appear natural to view any problem involving a continuous value as a regression problem, it can often be easier to instead transform it into a classification problem. This can be done by means of splitting the continuous range up into a finite set of *bins* and predicting what bin an instance belongs to, rather than what exact numeric value it should have. A common approach is to use two bins, one for values at or above a given threshold and one for values below a threshold. This approach of transforming a regression problem to a *binary classification problem* has seen widespread use for *in-situ monitoring* in additive manufacturing [3][11][16].

3.1.2 Accuracy Metrics for Binary Classification

In order to determine how well any sort of algorithm is performing, it needs to be evaluated. There is a large amount of different methods and metrics suitable for different situations depending on a range of factors. Since the thesis is looking at a binary classification problem, that is what this section will focus upon. For binary classification problems, a common way of viewing things is to assign one of the classes as the "positive" class and the other one as the "negative" class. The convention is to let the class that one wishes to identify as positive, for instance the occurrence of a defect in manufacturing or the presence of a disease in a medical test, and let the other case (often the normal situation) be the negative one, i.e. an object not having a defect in manufacturing or the absence of a disease in a medical test. Using this terminology, we get four possible outcomes from any binary classification:

- *True Positive* (denoted *TP* in formulas) is the case when a positive instance is correctly identified as positive
- *False Positive* (denoted *FP* in formulas) is the case when a negative instance is incorrectly identified as positive
- *False Negative* (denoted *FN* in formulas) is the case when a negative instance is incorrectly identified as positive
- *True Negative* (denoted *TN* in formulas) is the case when a negative instance is correctly identified as negative

Summarizing the occurrence for each of the cases and denoting them *TP*, *FP*, *FN* and *TN*, we can define some commonly used metrics:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

		Predicted Class	
		Positive	Negative
True Class	Positive	TP	FP
	Negative	FN	TN

Table 3.1: The four possible results in binary classification

Precision is a measurement of how often the classification is correct when identifying the positive class. If one considers a defect to be the positive case, precision is the rate of defective objects to all objects that were identified as defective. In manufacturing, where defective objects are scrapped, having a high precision is important to avoid material waste.

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Recall (also called *sensitivity* or *hit rate*) is a measurement of how often the classification catches a positive instance. Using the same example as for precision (i.e. a defect being the positive case), recall is the rate of all defective objects that were identified as defective. In manufacturing, having a high recall is important to avoid defective components accidentally being used.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3}$$

Accuracy is the rate of correctly classified instances. In a situation with an *unbalanced* data set (that is, a data set not containing an equal number of instances for all classes), accuracy quickly becomes unreliable as a measurement. If for instance one class occurs 90% of the time and the second only 10% of the time, always guessing that an instance belongs to the majority will result in an accuracy of 90%.

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.4}$$

F1-score combines precision and recall into a single measurement to weigh them together [17]. It is often used when a single metric is called for. However, since precision or recall are often not equally important (both in manufacturing and medical tests, it is usually more important to have a high recall rather than precision), F1-score is often not ideal [3].
A common tool for visualizing the trade-off between true positive rate and false positive rate is a *receiver operating characteristic curve* (ROC curve). A ROC curve displays the true positive rate as the false positive rate is varied. The better a classifier is, the closer it is to the top left corner. ROC curves are often drawn with a diagonal line down the middle: the performance of the dashed line can be achieved simply by guessing, which often is a useful benchmark. If a given situation calls for a numeric metric to evaluate a classification model, the area under the curve (called *ROC AUC*, short for Receiver Operating Characteristic Area Under Curve) can be measured [17]. An ideal classifier has a ROC AUC of 1, guessing has a ROC AUC of 0.5. Figure 3.2 shows an example of a ROC curve.

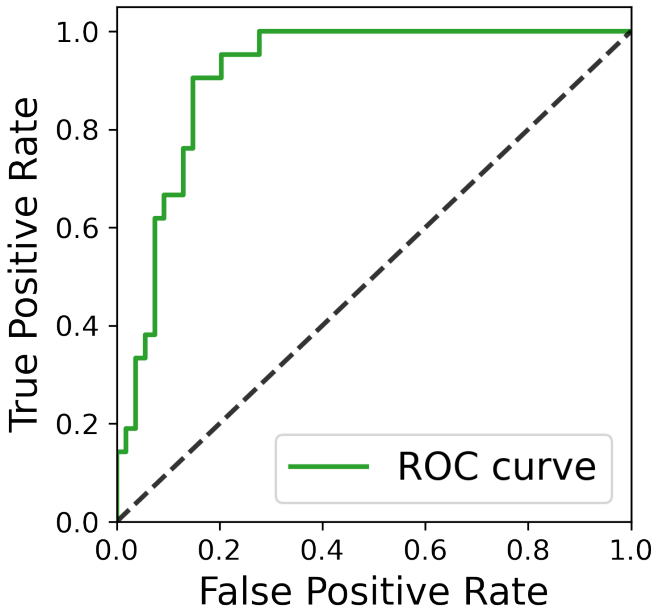


Figure 3.2: Example of a ROC curve

One final evaluation tool that is more specific towards manufacturing is the *Probability of Detection* (*POD*) curve. A POD curve displays the probability of detecting a defect (i.e. the recall) as the size of the defect varies. It can be a valuable tool since it provides information about how severely compromised an item can be, while still being flagged as OK. Although useful, creating a POD curve often requires rerunning experiments with different configurations. As a consequence, in the literature surveyed for this thesis only one paper used POD curves as part of the evaluation [3], and the paper in question was specifically about how to evaluate performance. Figure 3.3 shows an example of a POD curve.

Method	Benefits	Drawbacks
Accuracy	Easy to measure	Performs poorly on unbalanced data sets
F1-score	Works well with unbalanced data sets	Gives equal importance to precision and recall, which is often not desirable
ROC AUC	Accounts for both precision and recall in a single metric	Does not provide as much information about the trade-off between precision and recall as a ROC curve
ROC Curve	Visualizes the trade-off between precision and recall	Hard to optimize according to
POD Curve	Accounts for the magnitude of an instance	Hard to calculate, often requires re-running analysis with different settings

Table 3.2: Overview of classification evaluation methods

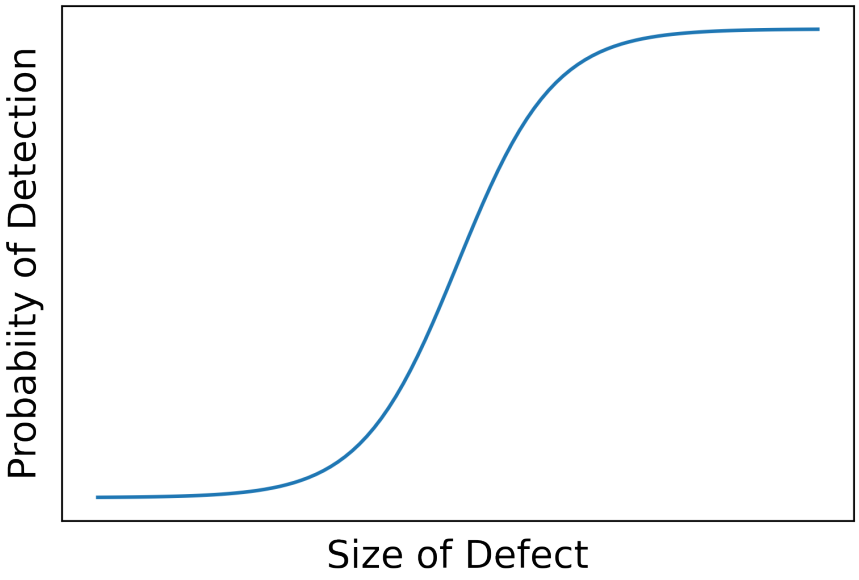


Figure 3.3: Example of a POD curve

Table 3.1 shows an overview of the evaluation methods.

3.2 Outlier Detection Methods

Outlier detection is the process of identifying data points in a set that are significantly different from the rest of the data points. For spatial data (such as the kind of images produced by in situ optical sensors), there are a special set of algorithms

called *spatial outlier detection* algorithms. Formally, a spatial data set is a data set that has one or more spatial dimensions describing the distance between data points, as well as potentially any number of non-spatial *attributes* [12]. In a data set of mountain peaks in a country, the latitude and longitude of the peak would be spatial dimensions, whereas the snow depth and number of visitors per year could be considered non-spatial dimensions. Outlier detection is different in spatial compared to other data because of a phenomenon known as *spatial autocorrelation*: that is, points which are close together in spatial dimensions often have similar non-spatial attributes. A peak in the north of Sweden covered in a deep layer of snow during the summer would be nothing out of the ordinary, whereas a peak in the south having any amount of snow would be rather different from peaks in the area. Similarly, if one peak had a high number of visitors compared to surrounding peaks, one might consider such a peak an outlier even if the number of visitors is lower compared to other peaks in the data set. Building on this intuition, we can define a *spatial outlier* as a data point that has a non-spatial attribute that is significantly different from its *neighbourhood*. The neighbourhood of a data point consists of the points that (by some distance function) are the closest to the point in the spatial dimensions.

The methods used in this thesis are explained in the following subsections.

3.2.1 Scatter Plot

A *scatter plot*, often used as a visualization tool, can be adapted to identify outliers. Consider the data points in image 3.4. Each point has a non-spatial attribute (the Y-axis) and a spatial location (the X-axis). By plotting each point with its attribute value on one axis and the average value of its neighbourhood (in this case, the point immediately below and above it in the spatial dimension), we get the plot shown in figure 3.5. The diagonal line is fitted to the points. Because the attribute value of each point in general is similar to that of its neighbours, the points tend to be close to the line, but there is some variance between how close they are. The distance from this fitted line to each point can be used as a metric to rank each point according to how much of an outlier it is. The further away from the line a point is, the stronger the outlier is. In the plot, points P, Q and S are furthest from the line: S because it has a significantly higher attribute value than its neighbourhood, and P and Q because they are next to S, resulting in them having lower attribute values than their neighbourhood [12].

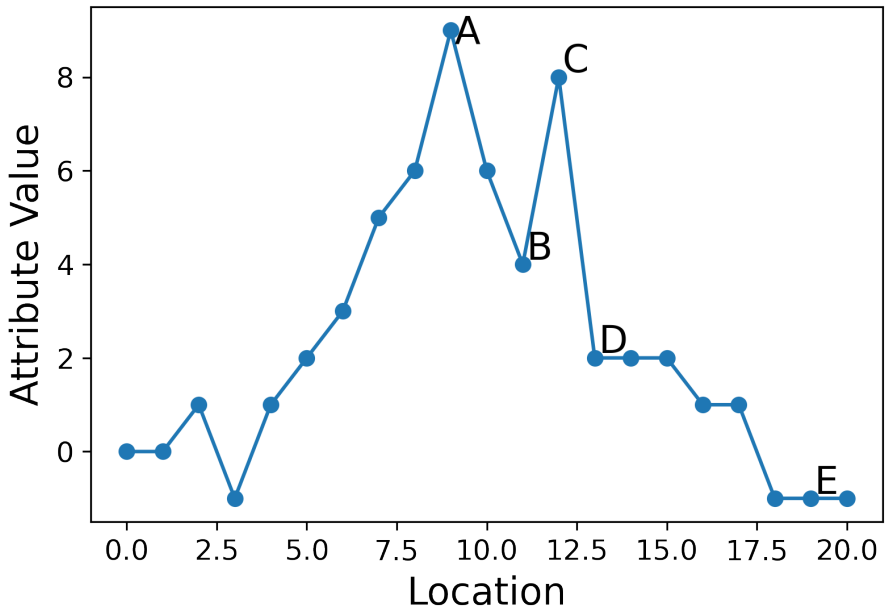


Figure 3.4: Example spatial data set with one spatial dimension and one non-spatial dimension

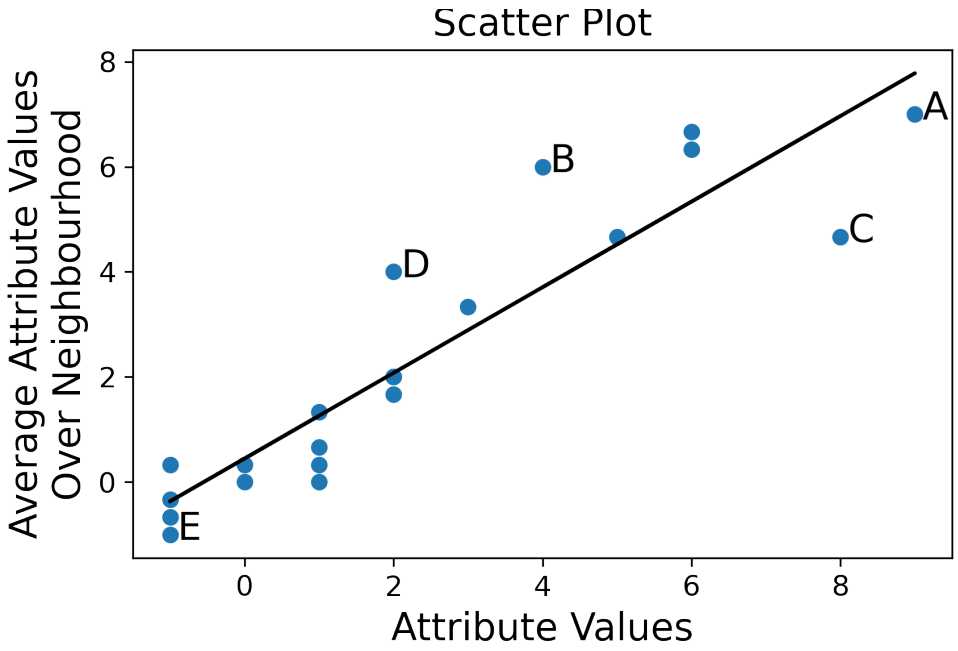


Figure 3.5: Scatter plot for the data set in figure 3.4. The further a point is from the fitted line, the more different it is compared to its neighbours

3.2.2 Moran Scatter Plot

A *Moran scatter plot* works similarly to a normal scatter plot, except instead of looking at the attribute value directly the Z-score of the value and the average Z-score of the neighbourhood is considered instead. The Z-score can be calculated according to equation 3.5, where μ_f is the mean of the attribute values and σ_f is the standard deviation of the attribute values [12]. The Z-score is also called the *standardized* value in machine learning terminology. Figure 3.6 shows a Moran scatter plot for the example data in figure 3.4.

$$Z[f(i)] = \frac{f(i) - \mu_f}{\sigma_f} \tag{3.5}$$

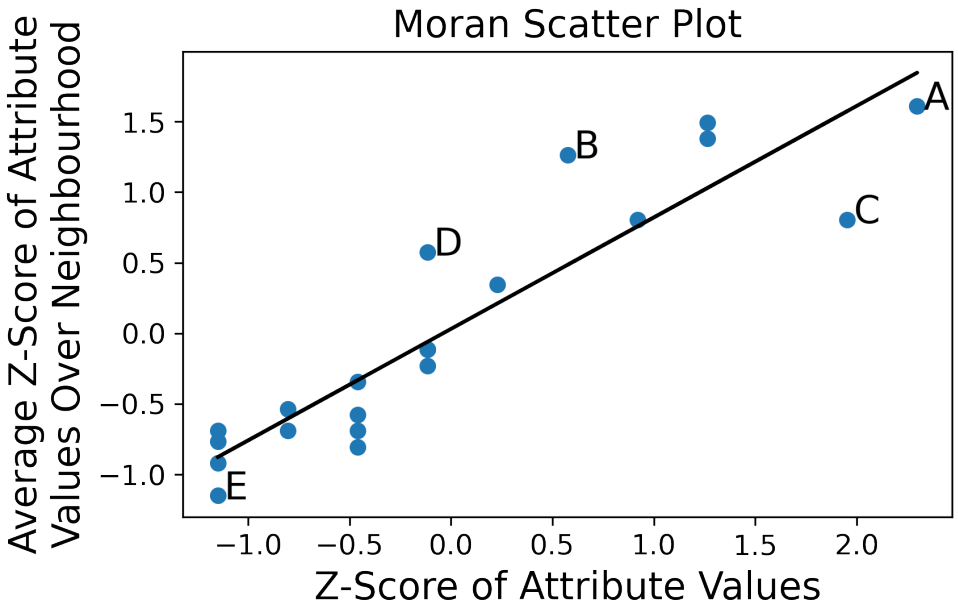


Figure 3.6: Moran scatter plot for the data set in 3.4. The further a point is from the fitted line, the more different it is compared to its neighbours

3.2.3 Spatial Statistic

Spatial statistic is a numeric method of identifying outliers. Similarly to Moran scatter plot, it uses normalization. There are two formulae that are relevant. First, we need to calculate the difference between the attribute value of x and the average in its neighbourhood. In equation 3.6 this difference is denoted *S(x)*. *f(x)* is the attribute value of point x and *m(x)* is the average of the attribute values for all neighbours of x (including x itself).

$$S(x) = [f(x) - m(x)] \tag{3.6}$$

The spatial statistic $Z_{s(x)}$ can then be calculated using formula 3.7, where μ_s is the average value of *S(x)* for all points in the data set and σ_s is the variance of *S(x)* for

all points in the data set. Using machine learning terminology, the spatial statistic is the Z-score of the difference between the attribute value of x and the average of its neighbourhood [12].

$$Z_{s(x)} = \left| \frac{S(x) - \mu_s}{\sigma_s} \right| \tag{3.7}$$

Figure 3.7 shows the spatial statistic for all data points in figure 3.4.

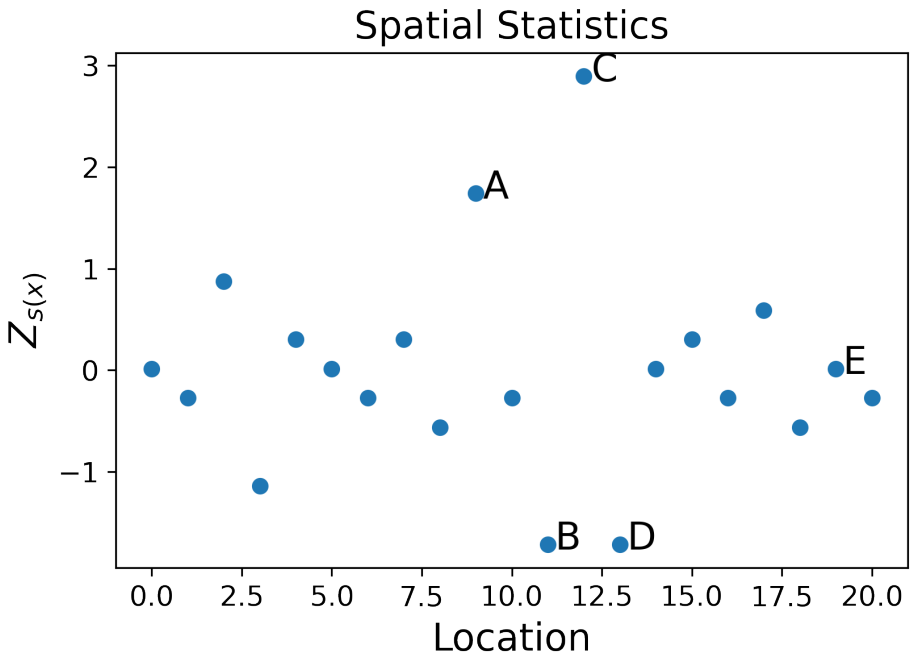


Figure 3.7: Spatial statistic values for the data points in figure 3.4. The further from 0 on the Y-axis a point is, the more different it is from its neighbours.

4

Problem Statement and State of the Art

This chapter begins by outlining the problem statement and scope of the thesis. It then provides an overview of the current state of the art of related in-situ monitoring research. It then presents a number of identified challenges for in-situ monitoring, along with opportunities for research.
This chapter assumes familiarity with additive manufacturing (covered in chapter 2) as well as machine learning and data mining(covered in chapter 3).

4.1 Thesis Problem Statement and Scope

An optical tomography setup reports the relative temperature of the powder bed, creating one image per layer during the build process. These images form a data stream, $S = s_1, s_2, \ldots, s_n$, with s_1 corresponding to the first layer of the constructed object and s_n the last layer. Figure 4.1 shows one such image for a build object. The relative temperature differs inside each individual object, with slimmer sections typically being warmer due to a combination of less thermal conductivity and less time to cool down compared to wider sections. This thesis identifies and quantifies "hot spots" in locations where they are unexpected to occur, which could be indicative of porosity due to worse thermal conductivity. Using this quantification, we then predict whether the object's porosity falls above or below a threshold, i.e. whether it is an object of acceptable or unacceptable porosity level.
Due to the availability of data, the scope of the thesis is limited to distinguishing objects with lack of fusion pores (i.e. insufficient energy input) from objects without them (i.e. normal energy input), thus not taking objects created with too high energy input into account. In addition, due to only having accurate porosity measurements for larger segments of objects since the data was acquired using optical microscopy, the thesis looks at classifying these larger segments instead of locating pores or porous areas.
The main metric used to evaluate the performance of the proposed method is the area under the receiver operating characteristic curve (ROC-AUC). This metric is used since it is a common metric for binary classification problems, it is not affected by class imbalance, and it has been used by previous research [3][13]. In addition, due to the timing constraints for in-situ monitoring (if it takes too long, construction of the next layer needs to be delayed), the execution speed will also be a factor in the evaluation of the proposed method.

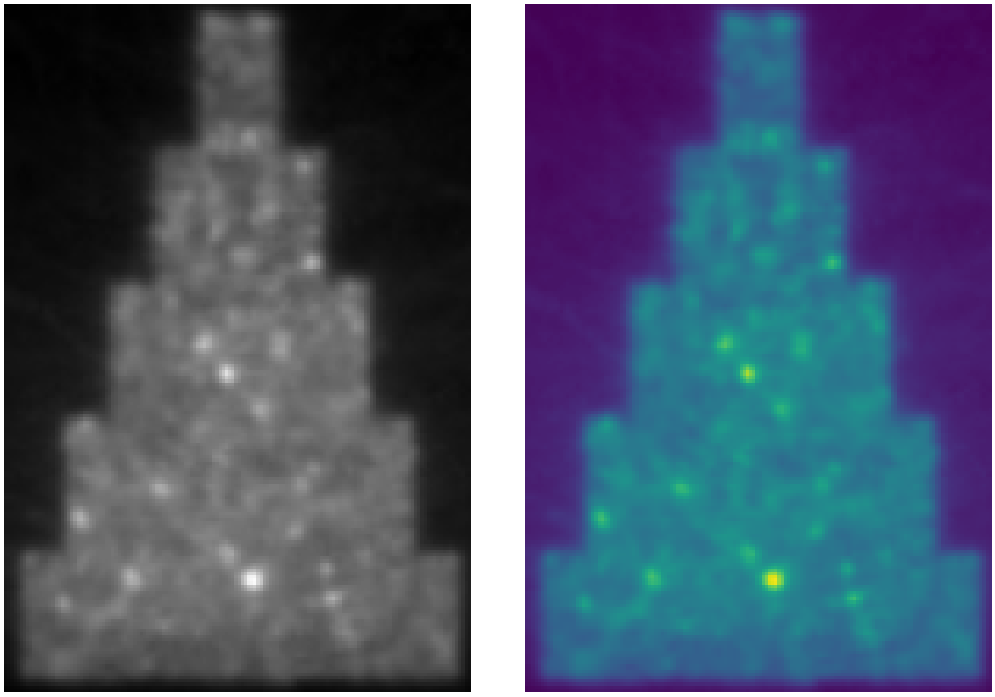


Figure 4.1: Example of an image of a build object. To the left, the image is shown in its original greyscale values. To the right, the same image is illustrated using pseudocolours

4.2 Overview of Previous Research

Multiple research projects have been conducted on this and related problems. Table 4.1 provides an overview of known research results relating to the topic of this thesis. Within the scope of the thesis, literature relating to two related research problems have been considered:

- Classification of entire layers or objects as porous or not
- Identifying locations of pores or porous areas

For both problems, similar sensors and methods have been applied, which is why both are included despite this thesis focusing on the first problem.
We can distinguish three parts in the approaches used:

1. Sensing, the process of acquiring data from the manufacturing process
2. Preprocessing, i.e. transforming the data into a form that can be used for classification
3. Analysis, i.e. drawing conclusions or making predictions from the processed data

4.2.1 Sensing methods

Multiple different types of sensors have been used to measure so-called *process signatures*, readings observed during the manufacturing process. Grasso and Colosimo provides an overview in their metastudy [9]. Although most methods used are optical

(either cameras or photodiodes), there has also been some research using ultrasound and vibration measurements. Since this thesis uses a camera setup, the rest of this section will focus on that.

When considering camera setups used, two types exist: *co-axical* and *off-axical* setups. Figure 4.2 illustrates the two setups.

4.2.1.1 Co-axical setups

In a co-axical setup, the sensor is in line with the laser, which is accomplished using semi-permeable mirrors. As a result, it senses whatever the laser is targeting. A lot of research using co-axical setups have used high-speed cameras (395 Hz in the paper by Zhang, Shunyu, and Yung 2019, [25], 100kHz in the paper by de Winton et al. [3]) to monitor how the melt-pool changes over time, which then has been used to identify the occurrence of pores and other defects [3][25][10].

4.2.1.2 Off-axical setups

In an off-axical setup, the camera monitors the whole powder bed from a static angle instead of just where the laser beam is at a given point. The sampling rate of the sensors used in this setup can vary greatly, from one sample per layer to several thousands per second (2.5 kHz in the paper by Estalaki et al. [13]). One benefit of off-axical setups is that they make it easy to create a three-dimensional representation of the object, since it's just a matter of aggregating all images of the same layer (commonly done by looking at the number of frames the grey value of each pixel is above a given threshold) into a single image and then layering the images on top of each other. This process is known as *optical tomography*. Although there exists research using cameras without any particular filter [8], most research has involved using a filter to observe the heat radiating from the object instead of radiation in the visible spectrum. As a result, the images have a single colour channel and are typically shown in greyscale or using *pseudocolours* (i.e. artificially added colours) to make the differences more visible.

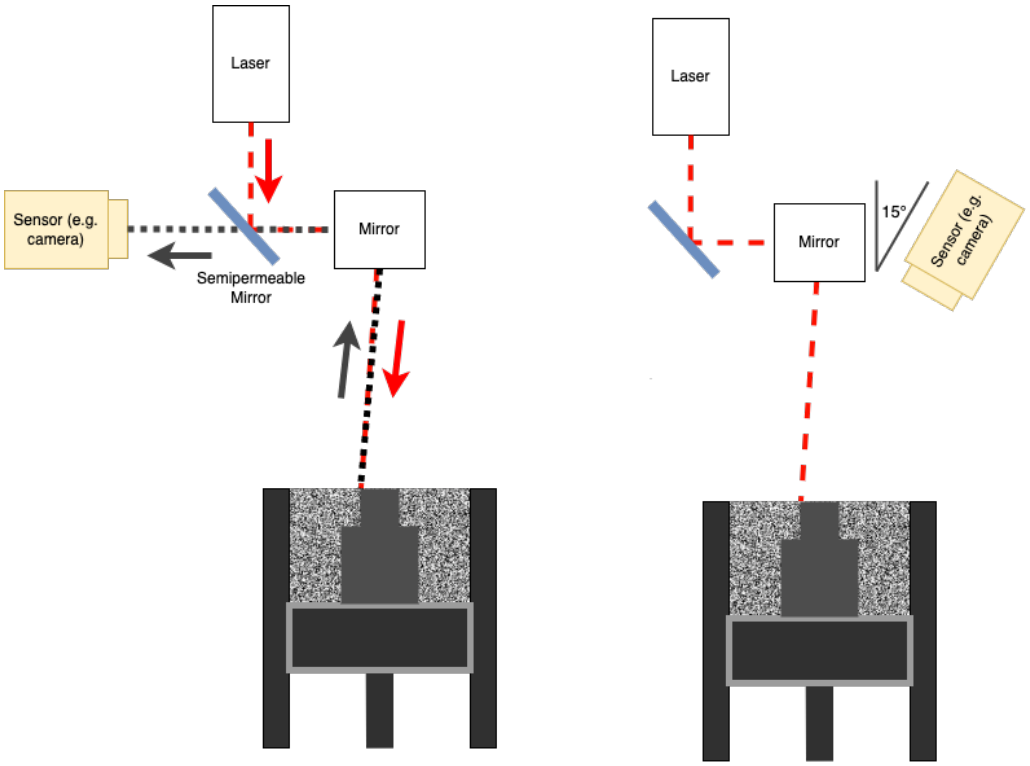


Figure 4.2: Illustration of a co-axical setup (left) and an off-axical setup (right)

Regardless of whether the camera is mounted co- or off-axically, different types of cameras with different filters can be used. Although cameras looking at the visual spectrum of light have been used in at least one study [8], a more common approach is to attempt to capture information about the thermal signatures from the process. This is done either using an infrared or near infrared camera, or a camera equipped with a bandpass filter adapted to capture the black body radiation omitted by the build object while filtering out the laser radiation. Table 4.1 shows the different types of cameras and setups used in the research.

4.2.2 Methods of Preprocessing and Analysis

Different approaches have been used for preprocessing and analysing sensor data, with different levels of complexity. On the one hand, in the research conducted by Zhang, Shunyu and Yung [10], the intensity values were just standardized (meaning the mean value of the intensity was subtracted and divided by the standard deviation) before being entered into a convolutional neural network. Gobert et al. [8] used the same method for preprocessing (i.e. standardization), but had a more complex analysis step. Due to them using multiple lighting setups with their camera, they used four support vector machines (a type of machine learning model) for different lighting settings. They then used the outputs from these support vector machines as input to another support vector machine, creating a so-called *ensemble model*.
In the study by Winton et al. [3], different methods of encoding were used. The most successful one was using computer vision algorithms to extract features of interest from each image, such as the number of spatters, the amount of spatter, the size of

Research Project	Problem considered	Type of sensor data	off/coaxical setup	Type of ground truth	Encoding method	Analysis method
This thesis	Classifying porous objects	Long exposure camera with narrow bandpass filter (900nm)	off	Optical microscopy	Various outlier detection based methods	K-Nearest Neighbour (could potentially expand)
de Winton et al 2021 [3]	Classifying porous objects	High speed camera with bandpass filter (700nm)	co	Optical microscopy	Quantification of key features using image processing algorithms	K-Nearest Neighbour
Estalaki et al 2022 [13]	Porosity prediction for a voxel	SWIR (short-wave infrared) camera	off	X-ray CT	TOT and maximum radiance for each voxel	Six different off-the-shelf ML algorithms
Lough et al 2022 [11]	Porosity prediction for a voxel	SWIR (short-wave infrared) camera	off	X-ray CT	TOT and maximum radiance for each voxel	Probability Map
Mohr et al 2020 [15]	Identifying porous areas of an object	MWIR (mid-wave infrared) tomography VIS-NIR (visible and near infrared) OT	off	X-ray CT	TOT and maximum radiance for each voxel	Qualitative
Gobert et al. 2018 [8]	Discontinuity localisation	DSLR camera (visible spectrum) with 8 different lighting conditions	off	X-ray CT	Pixel intensity	Support Vector Machine
Zhang, Shunyu, and Yung 2019 [24]	Pore localisation and porosity prediction for an area	High speed camera with narrow bandpass filter (532nm)	co	Optical microscopy and X-ray CT	Pixel intensity	Convolutional Neural Network

Table 4.1: Overview of existing research on in-situ monitoring using optical sensors

the spatter etc. These values were aggregated across multiple images following each other, and then used as input to a K-nearest neighbour model.
The research conducted by Mohr et al. [15], Lough et al. [11] and Estalaki et al. [13] used similar camera setups and preprocessing methods to each other. For each layer, they looked at the time each pixel was above a given temperature threshold (abbreviated *TOT*, time over threshold) as well as the maximum measured radiance of each pixel. Their methods of analysis did however differ: Mohr et al. did a qualitative assessment (i.e. they had an expert assess whether the results were useful or not). Lough et al. built upon their research by using probability maps (a statistical tool) for predicting porosity. However, Lough et al. only considered the readings of the pixel where they were trying to predict the porosity. This was addressed in the research by Estalaki et al., who took the values of the surrounding area into account. They looked at values in a grid stretching out up to three pixels in the X- and Y- dimension as well as three layers above and below, forming a grid of size 7x7x7. The values of these pixels were then used as input to six different off-the-shelf machine learning algorithms, out of which random forest performed the best.

4.2.3 Types of Ground Truth Data

Another aspect that sets the research apart is the type of ground truth data used. There are two methods used in the literature: *optical microscopy* and *X-ray* CT (computed tomography). When using an optical microscopy, the data is obtained by cutting an object apart, and taking a picture of the cross-section of the object. Figure 4.3 shows and example of such an image. This kind of data can be used to measure the porosity of an object, but since it only gives information about a cross-section the assumption is often made that this cross-section is representative of the object as a whole [19][3]. In addition, the method is only useful when trying to predict the porosity of larger areas, since individual pores occur in different locations in each layer. The main benefit of the approach is that it is fast and cheap. However, since the process involves destroying the object it cannot be used to ensure the quality of an object after it has been constructed, unlike X-ray CT, a *non-destructive techniques* (NDT). The drawback of X-ray CT is that it is more time-consuming and more expensive, since it requires more specialized equipment. However, X-ray CT can be used to obtain a three-dimensional representation of the object, making it possible to locate individual pores or calculate the porosity of much smaller areas.
Depending on the problem considered, one or the other tend to be used. For predicting porosity in a small area (such as individual voxels from an OT recreation) or locating discontinuities (such as pores), X-ray CT data is required. For predicting porosity across a whole object, optical microscopy have been used [3].

4.3 Challenges

From a computer science perspective, numerous challenges exist within the research field. These include:

Figure 4.3: Example of optical microscopy image. The white area is solid, the black area is pores.

- A lack of shared accuracy metrics. Different studies have used different metrics to evaluate their models, making it hard to compare results between studies. In a recent study by de Winton et al. [3], they proposed *ROC* (receiver operating characteristics) and *POD* (probability of detection) as metrics for future research.
- A lack of publicly available data sets. In each study encountered, the researchers have created their own data set by physically building and analysing new objects. This further complicates comparisons between studies: since everyone is working with their own data set, it is not possible to say with certainty how well one method would work on a different data set.
- Difference in materials, parameters and geometry makes it hard to generalize. Since changing these affect the input to the sensor used, machine learning models need to be retrained for each part constructed. Due to the volume of examples needed for training, this process can be costly [16].
- Closed loop control. Once defects can reliably be identified, there is a need to research how these can be corrected live.

4.4 Related Research in the field: Pore classification

For the interested reader, this section introduces some additional research in the field. The research presented here is not directly related to the work presented in

this thesis, but it is still relevant to the topic of this thesis.
As mentioned in section 2.3.1 about pores, there are three different kinds that occur under different circumstances and understanding what kind of pores exist in an object can provide useful information about what actions to take to improve quality. Snell et al. [22] investigated the use of unsupervised learning in the form of K-nearest neighbour clustering for understanding the different caricatures of pores and differentiating between the different types. They found that three-dimensional data was more useful than two-dimensional, and that it is harder to differentiate between gas pores and keyhole pores than lack-of-fusion pores and other pores.
In a recent paper by Schwerz and Nyborg [21], they used a convolutional neural network (CNN) to classify *optical tomography* images taken during the build process according to the types of pores in the image. The accuracy, precision and recall all were above 96%, which is a promising sign for future research applying convolutional neural networks on this and related problems.

5

Methods

The method used looks at predicting whether a build object is porous or not given a series of greyscale images taken from the surrounding layers. These images are used as input to the first step in a series of steps, where the result of each step is used as input to the next. This series of steps ends with a prediction of whether the images are from a porous object or not. Figure 5.1 provides an overview of the method proposed in the thesis. Each step of the method is described in more detail in the following sections.

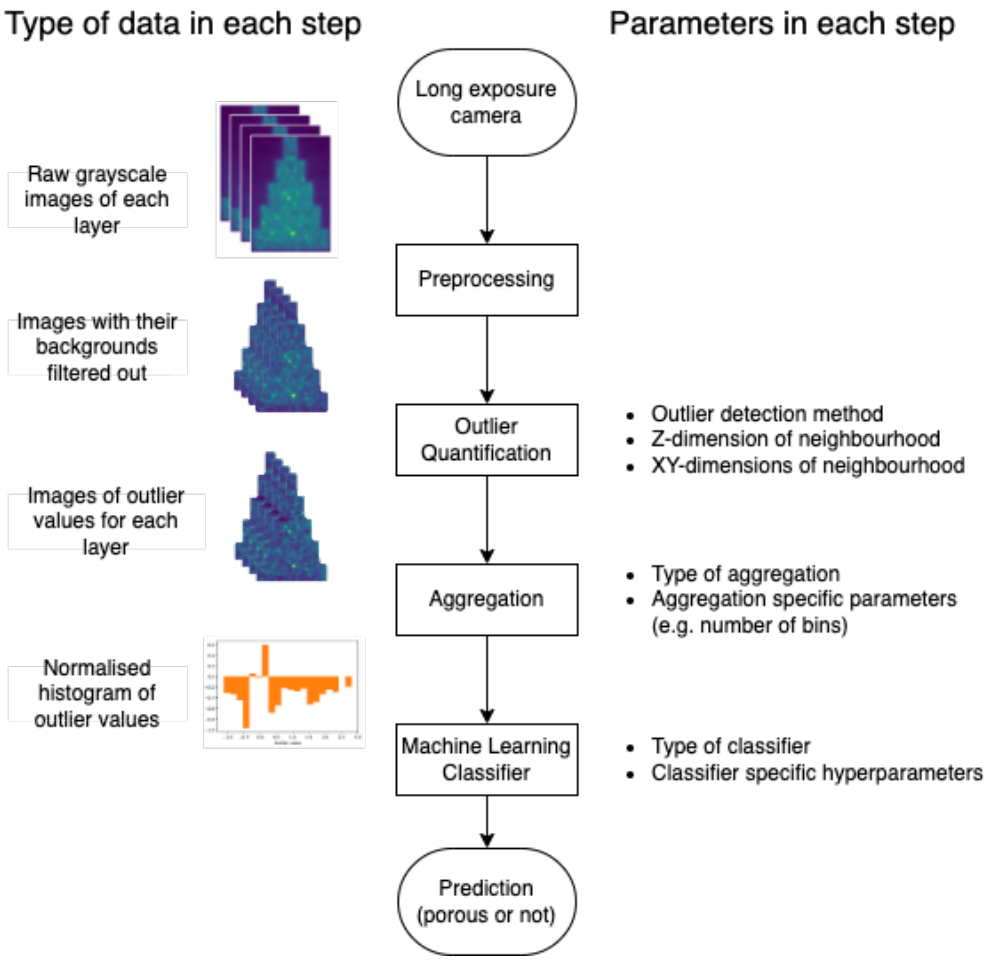
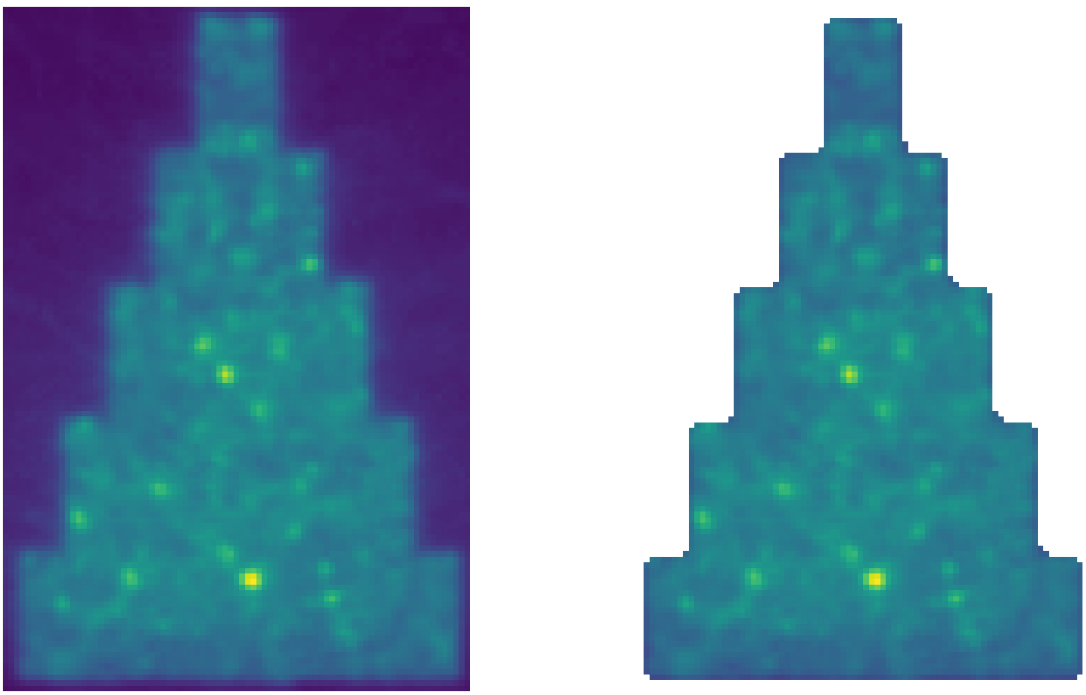


Figure 5.1: Overview of the method proposed in the thesis. The greyscale images are illustrated using a different colour scale to be consistent with the representation elsewhere in this thesis.

5.1 Preprocessing

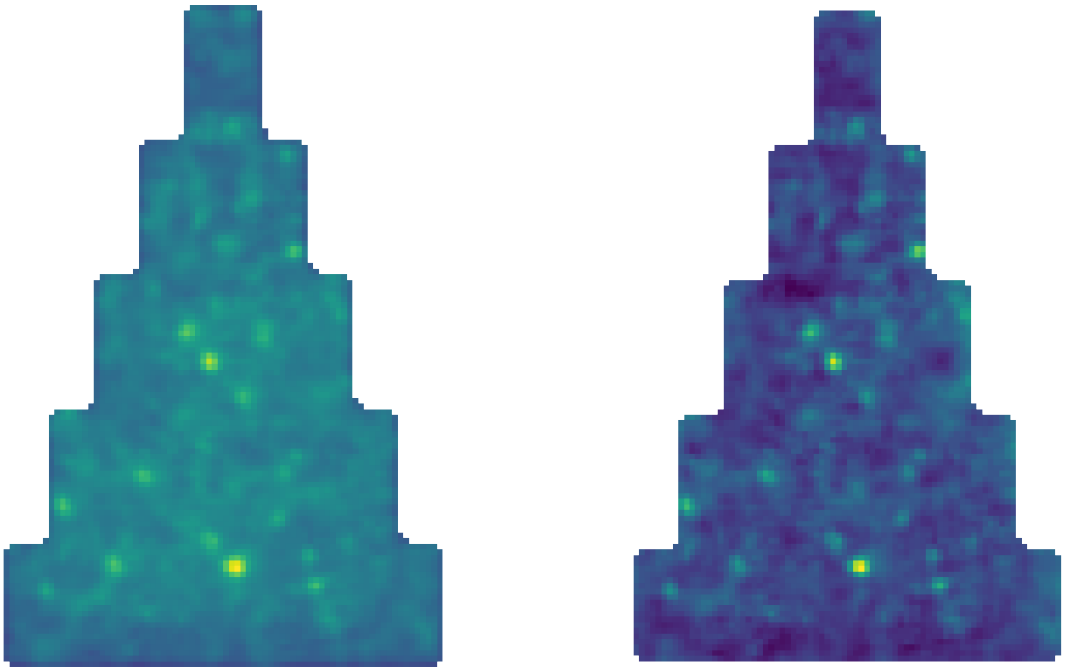
The first thing done to the images is a small amount of pre-processing, where the background is filtered out in order to avoid it being included in the analysis. Figure 5.2 shows an image before and after the background has been removed.



(a) Before background removal **(b)** After background removal

Figure 5.2: Background removal from the images during the preprocessing step

5.2 Outlier Quantification



(a) Input **(b)** Output

Figure 5.3: Visualization of an example input and output from spatial statistics

Once preprocessing is done, the result is forwarded to an outlier detection method used to quantify the severity and amount of outliers. Three different methods are investigated:

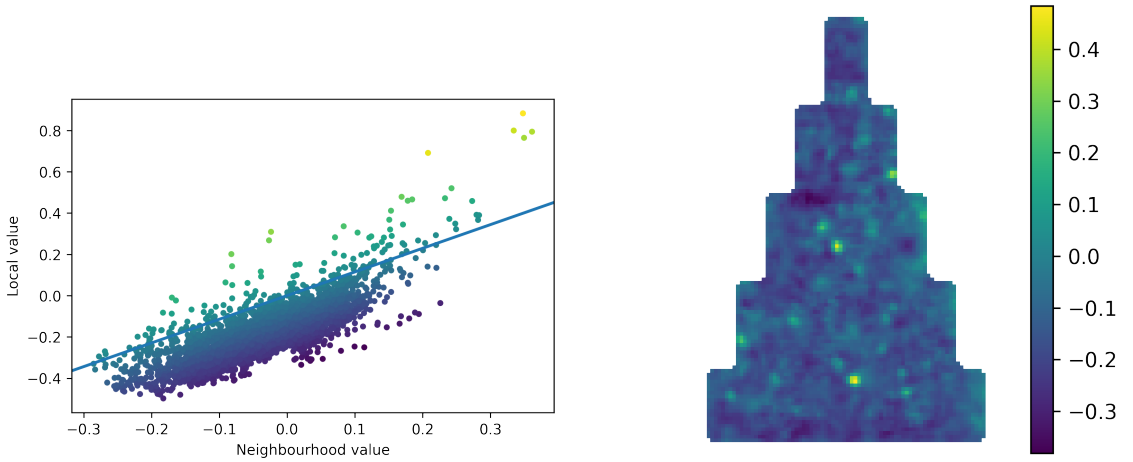
- Spatial Statistics
- Scatter plot
- Moran scatter plot

All methods work by comparing one central point to each point in its surrounding, also called its *neighbourhood*. The neighbourhood can be of any size, so it does not necessarily have to be limited to the points directly surrounding the central point. The neighbourhood can also be of any shape and use different weights depending on distance. For simplicity, a square neighbourhood is used in this thesis. The size of the neighbourhood in the X-, Y- and Z-directions are parameters of the outlier detection method.

The output of spatial statistics is a value for each point indicating to what extent it is an outlier. For the rest of this thesis, this value will be referred to as the *outlier value* of a pixel. Since each pixel has an outlier value, these can be visualized as an image (or series of images) with similar size to the original image. Figure 5.3 shows an example input and output for spatial statistics. The output is slightly smaller due to the neighbourhood at the edges being ill-defined: if a point has missing neighbours, how should that be accounted for? In this thesis, the points

with incomplete neighbourhoods are ignored, but other approaches exist as well.

Unlike spatial statistics, Moran scatter plot and scatter plot does not directly produce a numerical value for each point. They are visual tools for identifying outliers, and so there is a need to convert the result from these methods to a numeric representation. This can be done by fitting a line to the scatter plot and calculating the distance from each point to the line [12]. Figure 5.4 shows a Moran scatter plot with lines fitted to the data, as well as its corresponding image representation.



(a) Moran scatter plot of the input image with the background removed

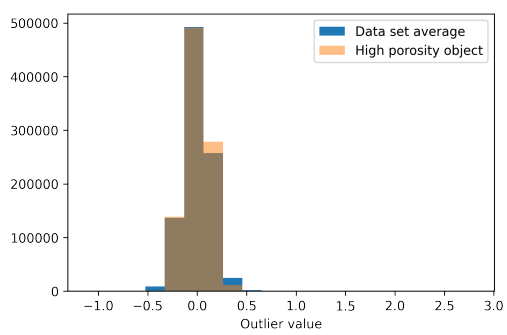
(b) Visualization of the output of Moran scatter plot

Figure 5.4: Moran scatter plot. The points are coloured based on their distance from the line. Each point in the scatter plot corresponds to a pixel in the image

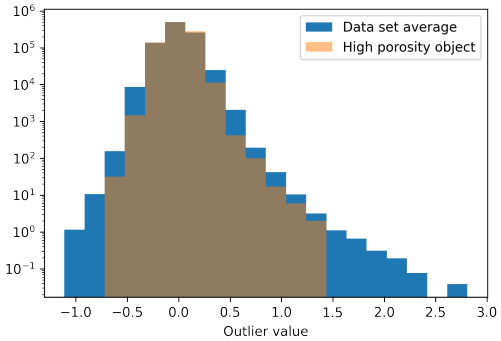
The time complexity for all the outlier detection methods discussed are $O(nml)$, where n is the number of pixels, m the number of layers considered as part of the neighbourhood and l the size of the neighbourhood. However, since only the next couple of layers affect the current one under normal operating conditions m can be viewed as a constant, reducing the time complexity to $O(nl)$.

5.3 Aggregation

Due to the high number of features from the outlier quantification (that is, each point corresponding to one feature), there is a need to reduce the dimensionality of the data. The approach used in this thesis to accomplish that is to convert the data to a histogram, although other options could be considered as well. Figure 5.5 compares the histogram of outlier values obtained using Moran scatter plot from one high-porosity object to 25 other objects. The blue is the average across all the 26 objects and orange is the high-porosity object. The dark orange/grey area is the overlap between the two.



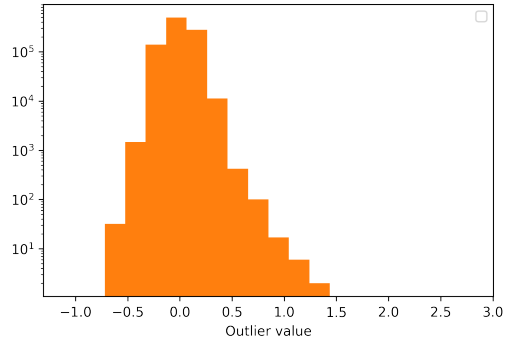
(a) Linear scale



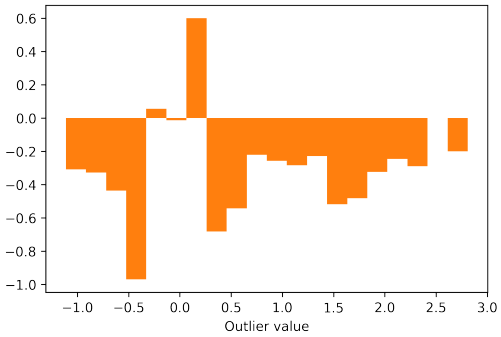
(b) Logarithmic scale

Figure 5.5: Histogram of outlier values obtained using Moran scatter plot from one high-porosity object compared with the average of 26 objects. The left plot is on a linear scale, while the right plot is on a logarithmic scale.

Due to the difference in scale between different features (i.e. bins of the histogram) and some machine learning models performing poorly on data with different scales, it is important to normalize the data. This can be done by subtracting the mean and dividing by the standard deviation. This is done for each feature separately. Figure 5.6 shows the histogram of the high porosity object from figure 5.5 before and after normalization.



(a) Before



(b) After

Figure 5.6: Histograms of the data points from the same high porosity object as in figure 5.5 before and after normalization.

5.4 Classification

The histogram is then used as input to a machine learning model. Any machine learning classifier could be used, for this thesis k-nearest neighbours classifier and decision tree are used due to their high degree of explainability.

A k-nearest neighbour classifier classifies an instance by looking at the majority class of the K nearest neighbours of the instance. Figure 5.7 shows an example for a data set having two features, X and Y.

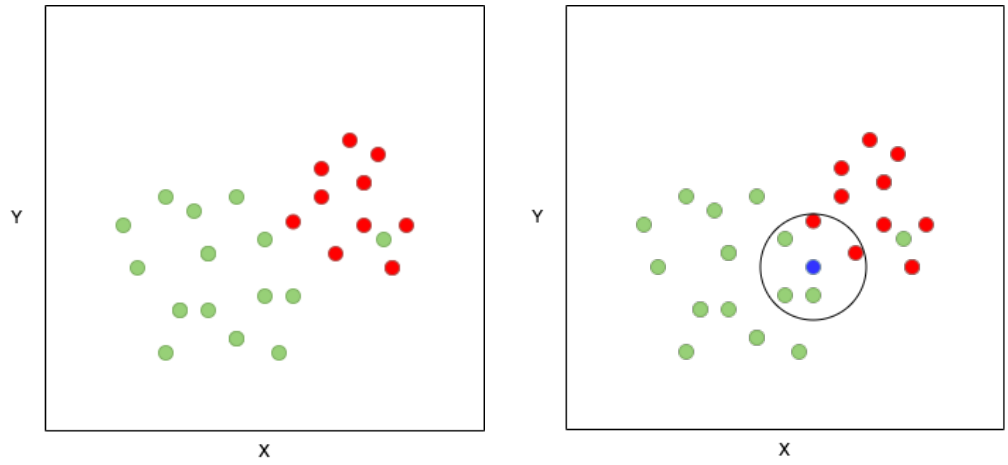


Figure 5.7: Example of how a K-nearest neighbour classifier works for a data set with two features, X and Y. On the left is the training dataset, with two classes (red and green). When the classifier is asked to predict the class of a test instance (blue) in the right image, it predicts the majority class among the K-closest instances from the training dataset. In this case, for $K = 5$, the majority class is green and so it predicts the test instance is green.

A decision tree is a tree-like structure where each leaf node represents a class label (such as porous/non-porous) and each internal node represent a condition to evaluate. Figure 5.8 shows an example of a decision tree.
Both of the classifiers were implemented using the Scikit-learn library [20].

5.5 Evaluation Methods and Metrics

For evaluating the classification results, this thesis uses the evaluation methods proposed by de Winton et al. [3]: *receiver operating characteristics (ROC)* and *probability of detection (POD)*. These are explained in more depth in chapter 3, but in short, due to the potentially high cost of false negatives (i.e. predicting that a component is fine when it is not) F1-score is not suitable in additive manufacturing. Furthermore, both de Winton et al. [3] and later research by Lough et al. [11] and Malakpour et al. [13] have used receiver operating characteristics. In cases when a single metric is needed, the area under the ROC curve is used *(ROC-AUC)*.
Probability of detection has seen less widespread use, but provides useful information about how well a classifier performs for different levels of severe defects. This is useful, since classifying an object with 0.8% porosity as non-porous is not as bad as classifying an object with 10% porosity as non-porous, something other common evaluation methods miss.
In terms of computational performance evaluation, since the recoating between each layer takes a few seconds, there is a "deadline" for the classifier to produce a result within that timeframe. This is not a hard deadline, but it is useful to meet it since otherwise construction time would increase.

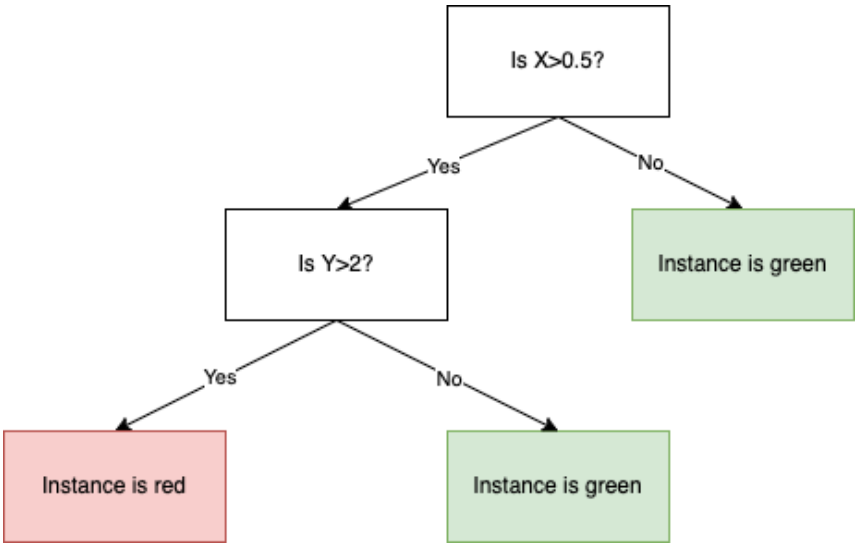


Figure 5.8: Example of a decision tree

6
Evaluation

This chapter explains how the method were evaluated and compares their performance. The chapter begins by outlining the experimental setup that was used for the evaluation, how the data was collected and how the ground truth labels were obtained. The result section then presents and compares the results of the different methods used.

6.1 Data Set Description

The method was evaluated on two sets of data consisting of 26 objects each. In the first set, referred to as the H-set (since the objects were cut horizontally), the build object is a five-tiered pyramid printed laying down. In the second set, referred to as the V-set (since the objects were cut vertically), the build object is a cube with two narrower sections on top, forming a shape similar to a house. Figure 6.1 shows the two shapes used in the data sets, and figure 6.2a shows the layout of the objects across the powder bed.

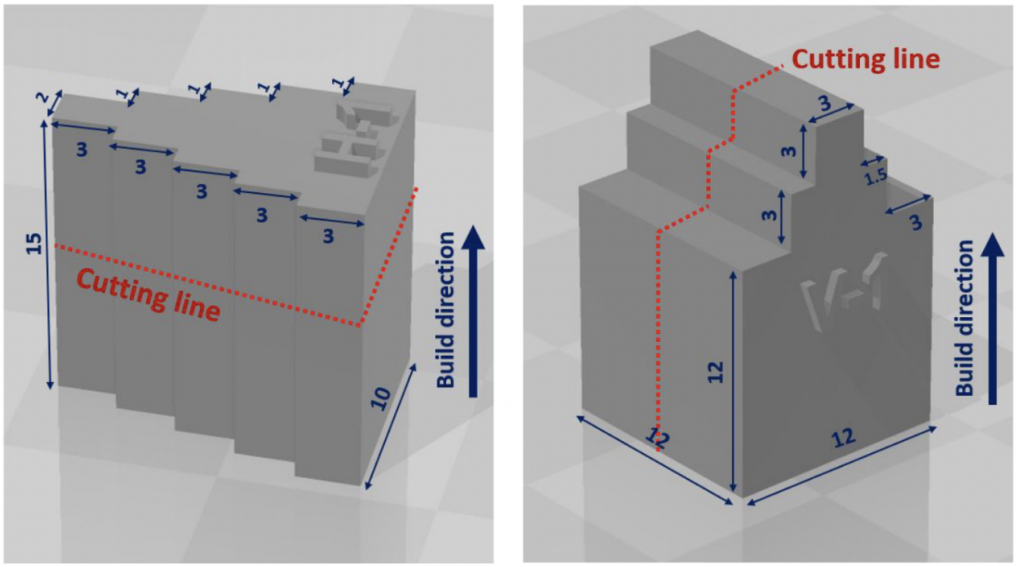
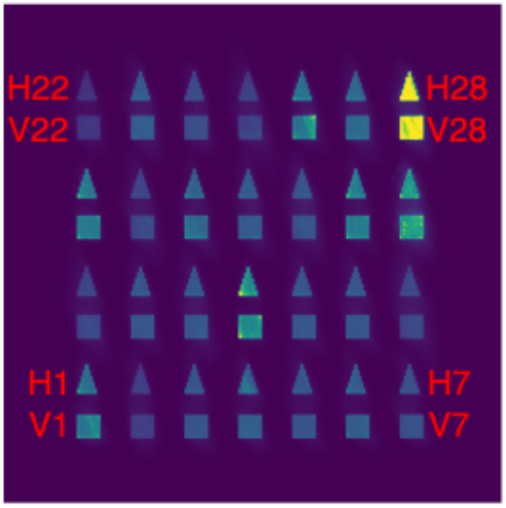


Figure 6.1: The two shapes constructed, numbers denoting distances in mm. On the left is an object from the H-set (cut horizontally) and on the right is an object from the V-set (cut vertically). Picture taken from [19]



(a) OT image of layer 92 during the construction process. The objects are numbered according to their position, from left to right and from bottom to top

(b) Example of an optical microscopy image of the cross-section of an object. Picture from object H8.

6.1.1 Test Object Construction

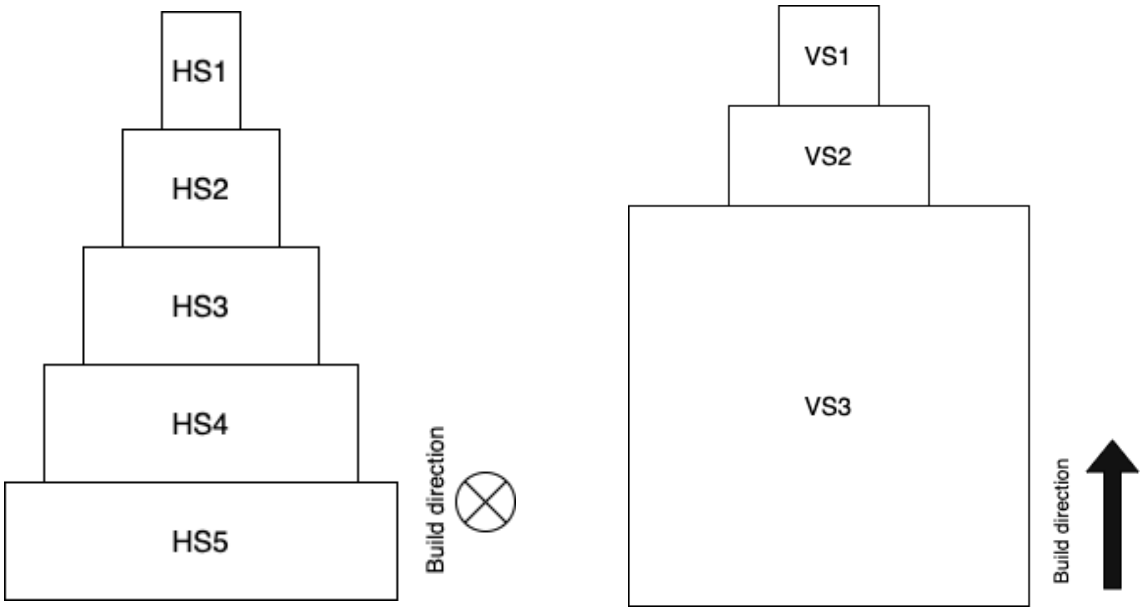
The objects in the two data sets were originally created for a different study investigating the effect of parameters on build quality [19]. Some of these parameter settings are outside what is considered as best practise and as a consequence, the result contains a mix of porous and solid objects. Table 6.1 shows the build settings used for the H-set and V-set, respectively. Note that two objects from each set (H21, H28, V21 and V28) are excluded from the evaluation: this was done since the build parameters resulted in a large amount of keyhole pores, and it was deemed infeasible to evaluate the method on these objects due to only having four objects of this type.
Each object was split into a number of segments: the objects in the H-set was split into five segments (one per "step" of the pyramid shape) and the objects in the V-set was split into three segments (the base, the middle and the top), producing a total of 130 data points in the H-set and 78 data points in the V-set. Figure 6.3 shows the segmentation.

6.1.2 Collection of Optical Tomography Data for Classification

The optical tomography (OT) images were collected using the built-in camera of the EOS M290 machine used for building the objects. The camera took one image per layer. The camera only uses one colour channel, resulting in greyscale images. Because of this, the *grey value* of a pixel is the brightness of the pixel, similar to how the red value of a pixel in a colour image is the amount of red in that pixel. The

Object	Laser energy (W)	Scan speed (m/s)	Hatch distance (µm)	Volumetric energy density
H1, V1	270	800	0.09	46.88
H2, V2	270	1200	0.13	21.63
H3, V3	300	1000	0.11	34.09
H4, V4	330	1200	0.09	38.19
H5, V5	300	1000	0.11	34.09
H6, V6	330	800	0.13	39.66
H7, V7	270	1200	0.09	31.25
H8, V8	330	1200	0.13	26.44
H9, V9	300	1000	0.11	34.09
H10, V10	270	800	0.13	32.45
H11, V11	330	800	0.09	57.29
H12, V12	300	1000	0.11	34.09
H13, V13	300	1000	0.11	34.09
H14, V14	300	1000	0.14	26.79
H15, V15	300	673	0.11	50.66
H16, V16	300	1327	0.11	25.69
H17, V17	349	1000	0.11	39.66
H18, V18	300	1000	0.11	34.09
H19, V19	251	1000	0.11	28.52
H20, V20	300	1000	0.08	46.88
H21, V21	250	664	0.08	58.83
H22, V22	250	1336	0.14	16.71
H23, V23	250	664	0.14	33.62
H24, V24	250	1336	0.08	29.24
H25, V25	350	1336	0.14	23.39
H26, V26	350	664	0.14	47.06
H27, V27	350	1336	0.08	40.93
H28, V28	350	664	0.08	82.36

Table 6.1: Overview of the parameters used when constructing each object in the H- and V-set. Objects H21 and H28 were excluded due to the high VED resulting in a lot of keyhole pores



(a) H-set **(b)** V-set

Figure 6.3: Schematic view of the segments in the two sets

higher the grey value (i.e. the brighter a given point appears), the warmer the point was. Note that these images are often coloured using a different colour map in the interest of making them easier to parse. In the case of this thesis, the *viridis* colour map is used, where the warmer points are coloured yellow and the cooler points are coloured dark blue.

6.1.3 Collection of Porosity Data Used as Ground Truth

The ground truth, i.e. the porosity values of each object, was collected by using an optical microscope to take a high resolution picture of the cross-section of each object. Figure 6.2b shows an example of the cross-section of an object, taken from object H8. The porosity was then calculated by measuring the amount of dark, hollow area to white, solid area in each segment of each object. Figure 6.4 shows the porosity of each object in relation to the average grey value of all pixels across all OT images of the object.

6.1.4 Parameters Used

For both evaluations, a range of parameters were evaluated:

- The **outlier detection methods** used were Moran scatter plot, scatter plot and spatial statistics. All of these methods work by assigning a numerical value to a point representing how much of an outlier it is according to the values in the neighbourhood (the area surrounding) the point. Since the Z-dimension represents the layers in each object whereas the X- and Y-dimensions represent distance inside the same layer, the Z-length was treated as a separate parameter from the X- and Y-length.

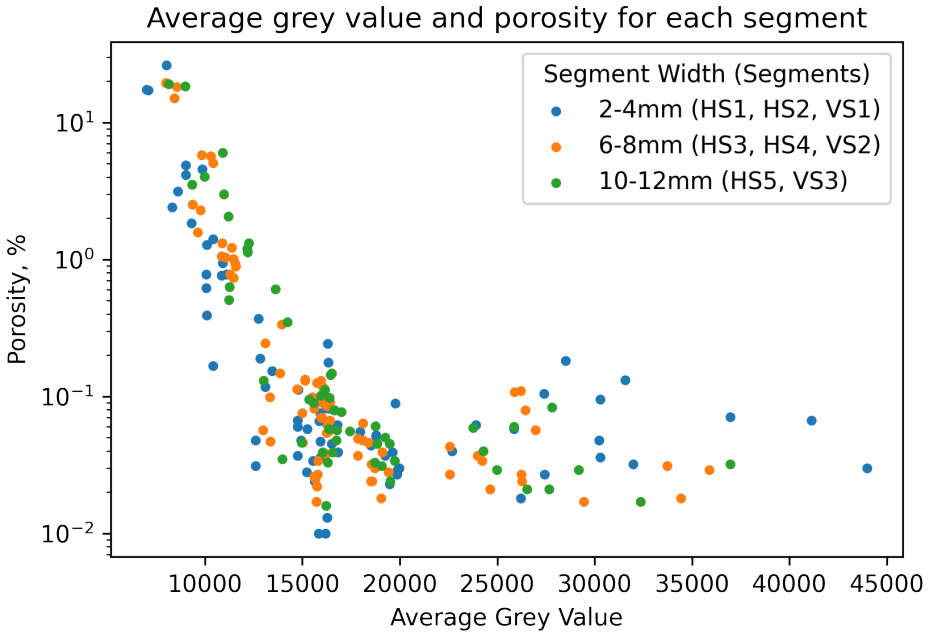


Figure 6.4: Porosity and average grey value for each segment in the two data sets. Each point represents one segment from one object, resulting in a total of 26*5=130 points from the H-set and 26*3=78 points from the V-set

- The **Z-length** of the neighbourhood is the depth of the neighbourhood, measured in number of layers. Each layer is 80 µm thick.
- The **X- and Y-length** of the neighbourhood is the size of the neighbourhood in the same layer, measured in pixels. The size of a pixel is approximately 125 µm x 125 µm.
- The only **type of aggregation** used was a histogram. Other types of aggregation are possible, for instance picking values at given percentiles, but were not used in this thesis.
- The **number of bins** is specific to the aggregation method chosen (histogram), and is the number of bins used in the histogram.
- The **type of classifier** is the machine learning model used to classify the data. The models used were decision tree and k-nearest neighbour classifier.
- **Hyperparameters** are the parameters of the classifiers and vary depending on the classifier. For the decision tree, the class weights were balanced to be inversely proportional to the frequency of the two classes (i.e. porous and non-porous), and the maximum depth was altered between 10, 14 and 22. For the k-nearest neighbour classifier, the number of neighbours was altered between 5, 7 and 9.

Table 6.2 provides an overview of the parameter settings used. All combinations of settings were attempted, resulting in 288 different combinations of settings.

Parameter	Settings	Description
Outlier detection method	Moran scatter plot, scatter plot, spatial statistics	Outlier detection method used
Z-size of neighbourhood	1, 3, 5, 7	Length of the neighbourhood in the Z-dimension measured in number of layers
XY-size of neighbourhood	3, 5, 7	Length of the neighbourhood in the X- and Y-dimensions measured in pixels
Type of aggregation	Histogram	Type of aggregation used to combine the values produced by the outlier detection method
Number of bins	5, 10, 20, 40	Number of bins to use when calculating the histogram
Type of classifier	Decision tree, k-nearest neighbour classifier	Machine learning model used to produce classifications from the aggregated values
Class weights (decision tree only)	Balanced	Weights used for each class when building the decision tree
Max-depth (decision tree only)	10, 14, 22	The maximum depth of the decision tree
k (k-nearest neighbour classifier only)	5, 7, 11	The number of neighbours considered

Table 6.2: Parameters settings investigated

6.2 Results

Two different evaluations were done. In the first evaluation, the H-set was split into a training and a test set. In the second evaluation, the H-set was used for training and the V-set for testing in order to see how well the model could generalize to a different geometry.
Since it is of interest to understand what role the surrounding area plays when predicting porosity, the neighbourhood Z-length and XY-length was varied, whereas the number of bins in the histogram and the hyperparameters of the classifiers were determined through 5-fold cross-validation.
Three different thresholds for porosity were investigated: 0.5%, 0.25% and 0.1%. The 0.5% and 0.25% thresholds are the same as the ones used by de Winton et al. [3], and the 0.1% threshold was included in agreement with Zhouer Chen, a researcher in material sciences at Chalmers University of Technology, as a lower limit of what is interesting to look at. According to Chen, although pores can still impact the quality even if the porosity is under 0.1%, it is more interesting to look at pore size and location at that scale, something that is not possible with the ground truth data used in this thesis.
In order to provide some indication of the performance of the method, there is a need for a baseline classifier. As shown in figure 6.4, there is a negative correlation between the average grey value across all OT images of an object and the porosity (i.e. for low average grey value, the porosity is high and vice versa). Because of this, a decision tree with a depth of 1 only considering the average grey value was used to provide a baseline to compare against for the outlier detection method setups used. The following section provides an overview of the results. A complete set of tables for all settings can be found in appendix A.

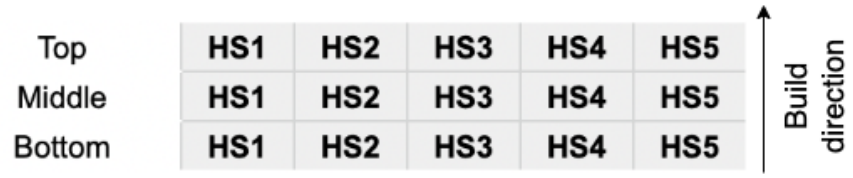
6.2.1 Evaluation on the H-set

Since the H-set on its own only consisted of 130 instances (26 objects with five segments each), the OT data of the objects were split into three parts to increase the number of instances. Each OT representation was split horizontally, such that the bottom layers, the middle layers and the top layers were kept separate, effectively producing three different, thinner instances of the original, thicker object. Figure 6.5 shows how the OT data for one object was split.
In the following subsections, the results for each threshold are presented in more detail.

6.2.1.1 Evaluation for Threshold at 0.50% Porosity

For the 0.50% porosity threshold, the setting with the highest ROC-AUC was the scatter plot with a k-nearest neighbour classifier, neighbourhood Z-length of 1 and XY-length of 3, which had a ROC-AUC of 0.989. Figure 6.6a shows the ROC curve for the three best settings according to ROC-AUC, as well as the baseline classifier. As can be seen in the ROC, depending on the desired true positive rate (TPR) or false positive rate (FPR), three different classifiers are ideal, with one of them being the baseline classifier. Figure 6.7 shows the confusion matrices for the four classifiers

(a) The original instances



(b) The instances after splitting

Figure 6.5: Schematic view of the splitting of the build objects during the H-set evaluation, as seen from the side. Figure 6.3a shows the segments as seen from above

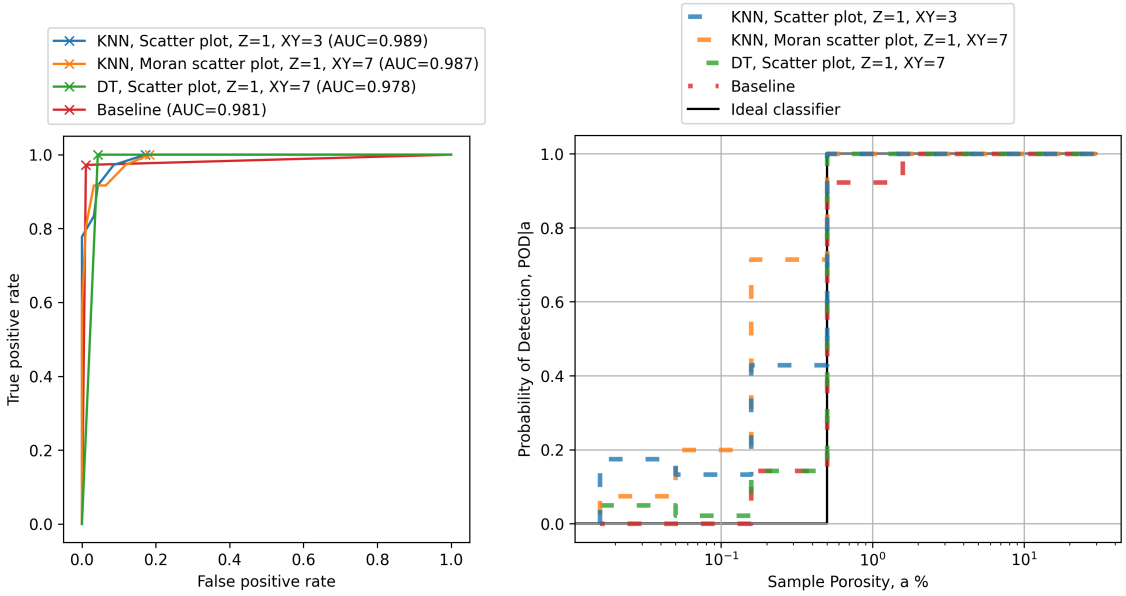
at the crosses in the ROC curve, and figure 6.6b shows the POD plot. All results can be found in table A.2 in the appendix.

6.2.1.2 Evaluation for Threshold at 0.25% Porosity

For the 0.25% porosity threshold, the best score was achieved by the k-nearest neighbour, scatter plot, neighbourhood Z-length of 3 and XY-length of 5 setting, that had a ROC-AUC of 0.975. Figure 6.8a shows the ROC curve for the three best settings according to ROC-AUC, as well as the baseline classifier. As can be seen in the ROC curve, similarly to the 0.5% threshold the best classifier depends on the trade-off between TPR and FPR, with three different settings being ideal at different intervals, one of them again being the baseline classifier. Figure 6.9 shows the confusion matrices for the four classifiers at the crosses in the ROC curve, and figure 6.8b shows the POD plot. Table A.3 in the appendix contains all results at the threshold.

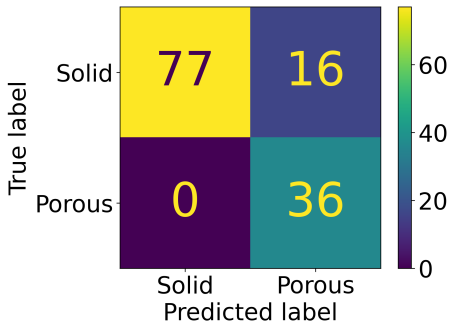
6.2.1.3 Evaluation for Threshold at 0.1% Porosity

Finally, at the 0.1% porosity threshold the best scoring setting was the k-nearest neighbour, Moran scatter plot, neighbourhood Z-length 1 and XY-length 7 setting, which had a ROC-AUC of 0.940. Figure 6.10a shows the ROC curve for the three best settings according to ROC-AUC, as well as the baseline classifier. Unlike at the two other thresholds, at this one any of the four classifiers may be the best depending on the TPR/FPR trade-off. Figure 6.11 shows the confusion matrices for the four classifiers at the crosses in the ROC curve, and figure 6.10b shows the POD plot. Table A.4 of the appendix shows all results.

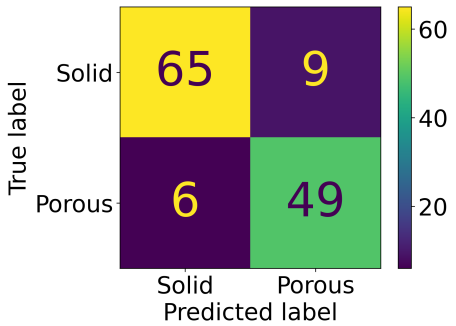


(a) ROC plot **(b)** POD plot

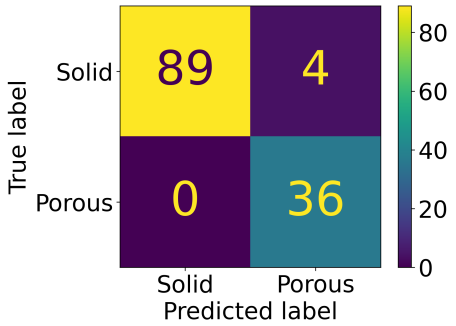
Figure 6.6: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.50% porosity threshold of the H-set evaluation. The crosses mark the location of the confusion matrices and pod plot



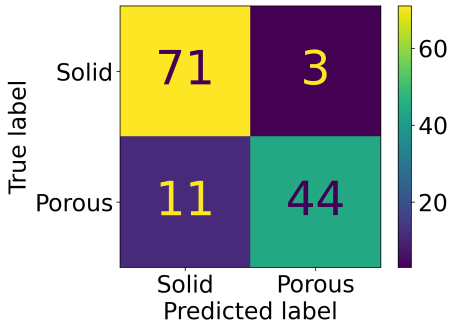
(a) KNN, Scatter plot, Z=1, XY=3



(b) KNN, Moran scatter plot, Z=1, XY=7

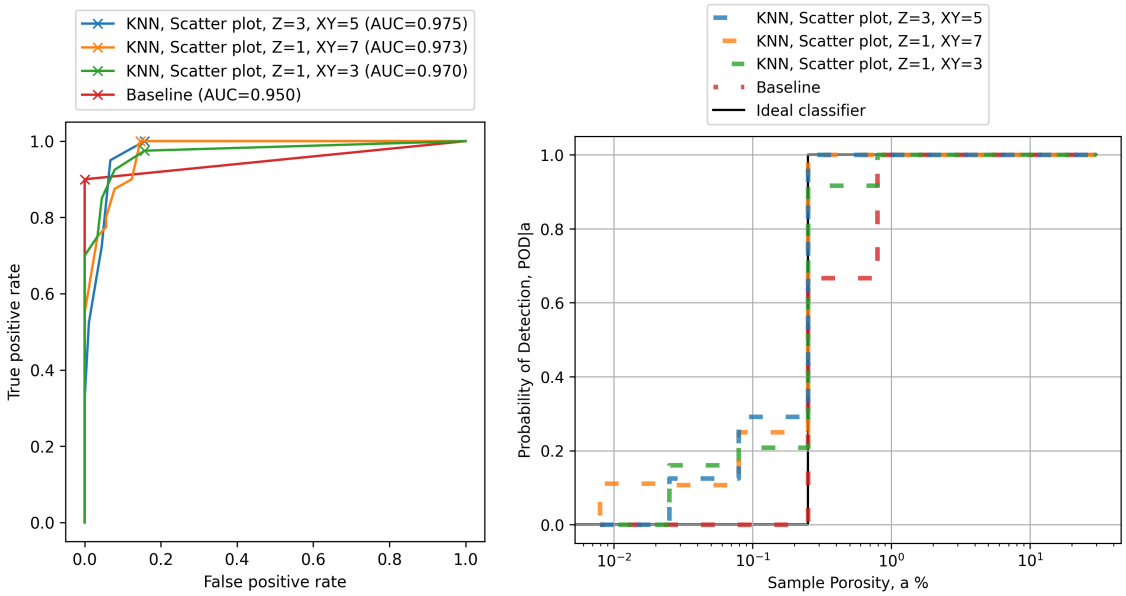


(c) KNN, Scatter plot, Z=1, XY=7



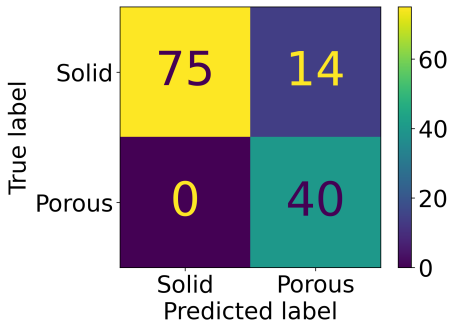
(d) Baseline

Figure 6.7: Confusion matrices for the three best settings and the baseline classifier for the 0.50% porosity threshold of the H-set evaluation

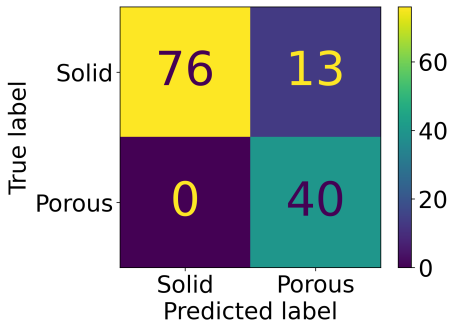


(a) ROC plot **(b)** POD plot

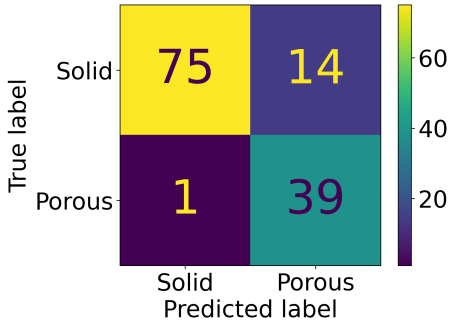
Figure 6.8: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.25% porosity threshold of the H-set evaluation. The crosses mark the location of the confusion matrices and pod plot



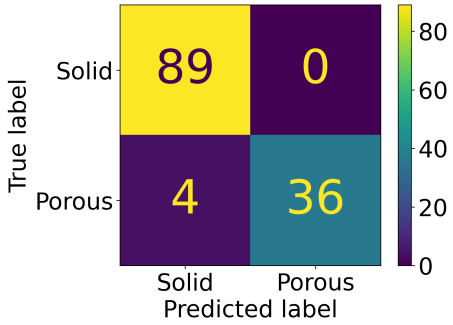
(a) KNN, Scatter plot, Z=3, XY=5



(b) KNN, Scatter plot, Z=1, XY=7

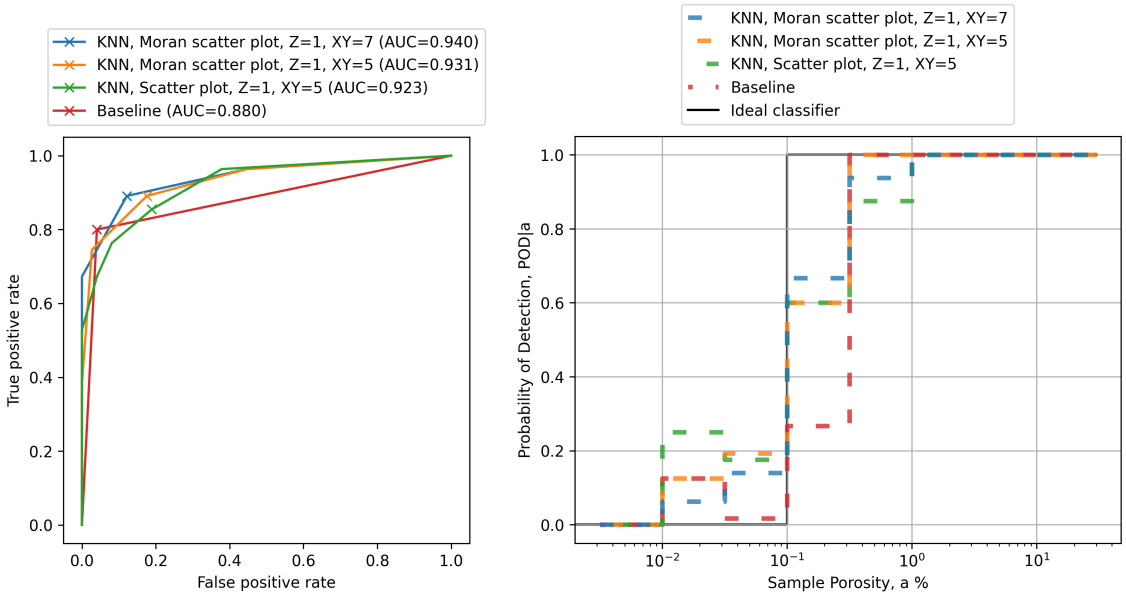


(c) KNN, Scatter plot, Z=1, XY=3



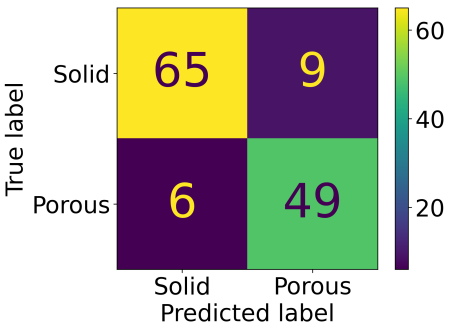
(d) Baseline

Figure 6.9: Confusion matrices for the three best settings and the baseline classifier for the 0.25% porosity threshold of the H-set evaluation

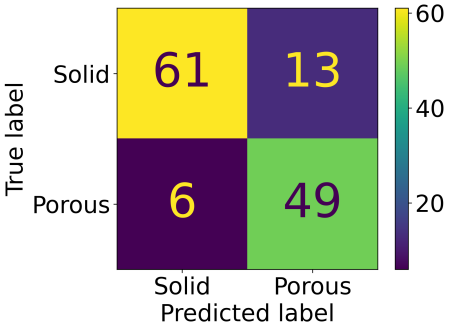


(a) ROC plot **(b)** POD plot

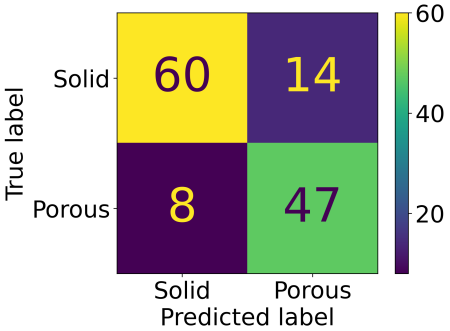
Figure 6.10: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.10% porosity threshold of the H-set evaluation. The crosses mark the location of the confusion matrices and pod plot



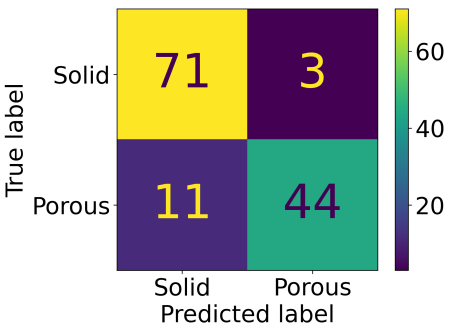
(a) KNN, Moran scatter plot, Z=1, XY=7



(b) KNN, Moran scatter plot, Z=1, XY=5

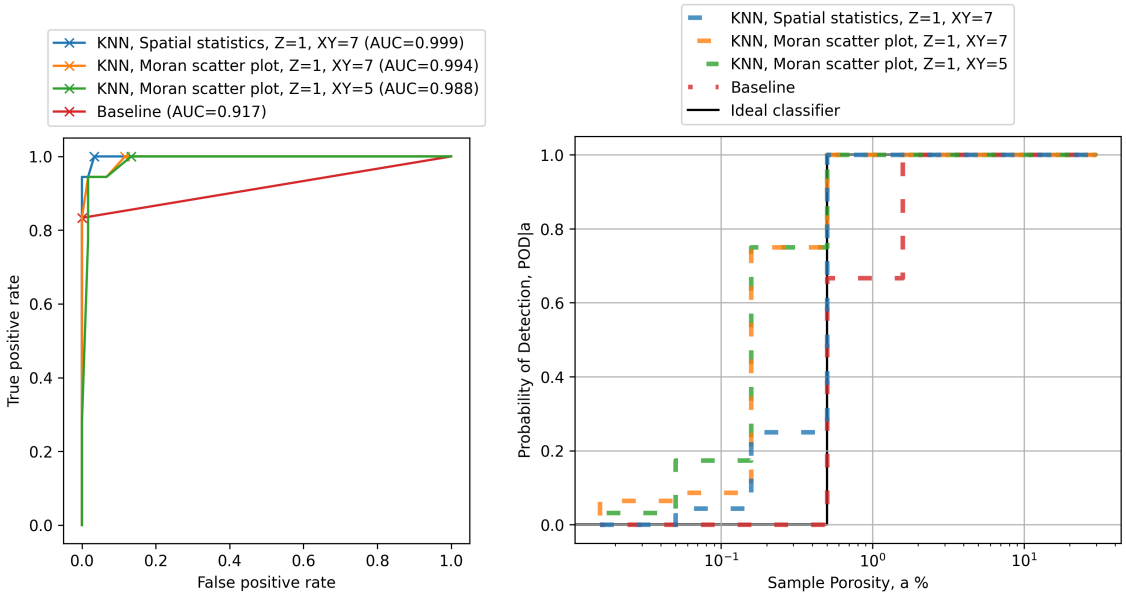


(c) KNN, Scatter plot, Z=1, XY=5



(d) Baseline

Figure 6.11: Confusion matrices for the three best settings and the baseline classifier for the 0.10% porosity threshold of the H-set evaluation



(a) ROC plot **(b)** POD plot

Figure 6.12: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.50% porosity threshold of the V-set evaluation. The crosses mark the location of the confusion matrices and pod plot

6.2.2 Evaluation on the V-set

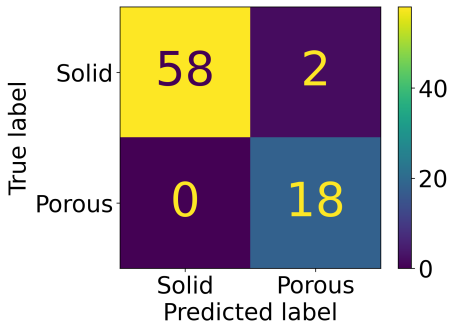
For the evaluation on the V-set, the objects in the H-set were used for training and the objects in the V-set were used for testing in order to see how well the method could generalize between different geometries. Unlike the evaluation on the H-set, the objects were not separated into multiple instances but kept whole, meaning the training set had 130 instances (five segments per object and 26 objects) and the test set had 78 instances (three segments per object, 26 objects).

Table A.5 provides a summary of the best settings for each classifier and outlier detection method at each threshold, as well as the performance of the baseline classifier. In the following subsections, the results for each threshold are presented in more detail.

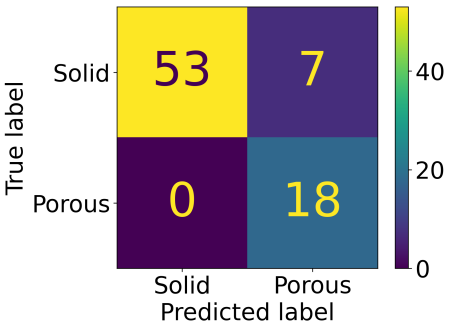
6.2.2.1 Evaluation for Threshold at 0.50% Porosity

For the 0.50% porosity threshold, the best ROC-AUC was achieved using the k-nearest neighbour classifier with spatial statistics as the outlier detection method and a neighbourhood Z-length of 1 layer and an XY-length of 7, for which the ROC-AUC was 0.999. Multiple other settings also outperform the baseline, although none of the scatter plot settings. All results can be found in table A.6 in the appendix.

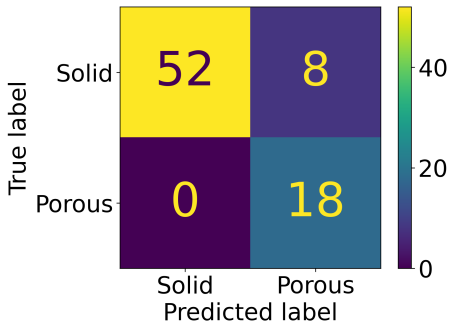
Figure 6.12 shows a ROC and POD plot and figure 6.13 shows the confusion matrices for the top three settings, as well as the baseline classifier. Unlike the H-evaluation, now there's clearly one dominant setting that regardless of FPR/TPR trade-off performs at least as well as the other settings.



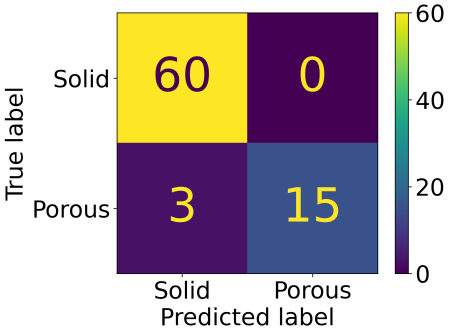
(a) KNN, Spatial statistics, Z=1, XY=7



(b) KNN, Moran scatter plot, Z=1, XY=7

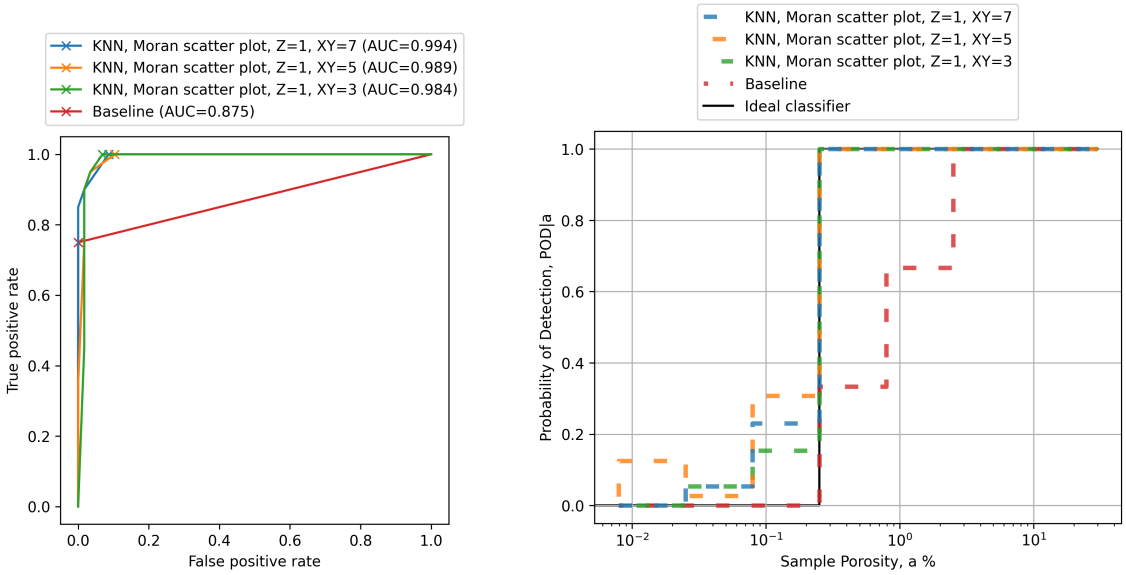


(c) KNN, Moran scatter plot, Z=1, XY=5



(d) Baseline

Figure 6.13: Confusion matrices for the three best settings and the baseline classifier for the 0.50% porosity threshold of the V-set evaluation



(a) ROC plot **(b)** POD plot

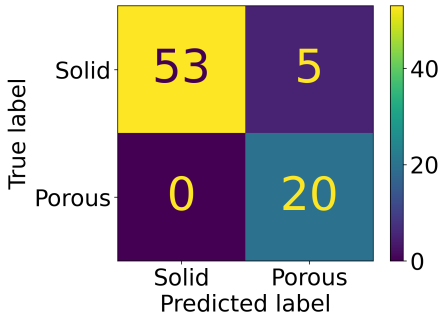
Figure 6.14: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.25% porosity threshold of the V-set evaluation. The crosses mark the location of the confusion matrices and pod plot

6.2.2.2 Evaluation for Threshold at 0.25% Porosity

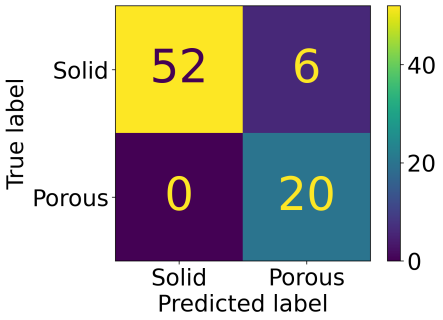
For the 0.25% porosity threshold, the best ROC-AUC was achieved using the k-nearest neighbour classifier with Moran scatter plot as the outlier detection method and a neighbourhood Z-length of 1 layer and XY-length of 7. The results for numerous settings are identical to at the 0.5% threshold, likely due to the two data sets having very few instances with a porosity in this range: for the test set, it is 2 out of 78 instances (2.5%) whereas for the training set it is 4.6%. One of the settings (KNN, Moran scatter plot, Z=1, XY=7) performed better at the 0.25% threshold than the 0.5% threshold due to it classifying the two instances in the 0.25%-0.5% porosity range as porous for both thresholds. All results are shown in table A.7 in the appendix.
Figure 6.14 shows a ROC curve and POD plot and figure 6.15 shows the confusion matrices for the baseline classifier and the top three settings.

6.2.2.3 Evaluation for Threshold at 0.1% Porosity

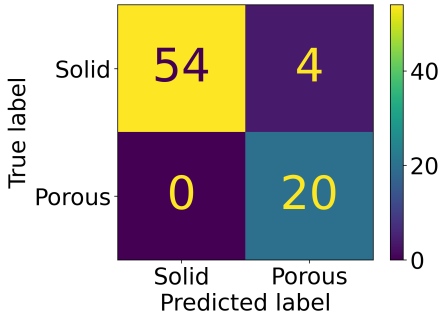
Finally, at the 0.1% porosity threshold the best result was achieved using the k-nearest neighbour classifier with Moran scatter plot as the outlier detection method and a neighbourhood Z-length of 1 layer and XY-length of 7, which had a ROC-AUC of 0.904. There are numerous settings that outperformed the baseline classifier (which had a ROC-AUC of 0.819) at this threshold. However, as for the other thresholds in the V-set evaluation, none of the settings using the decision tree classifier or scatter plot for outlier detection outperformed the baseline classifier. The results for all settings are shown in table A.8 in the appendix.



(a) KNN, Moran scatter plot, Z=1, XY=7



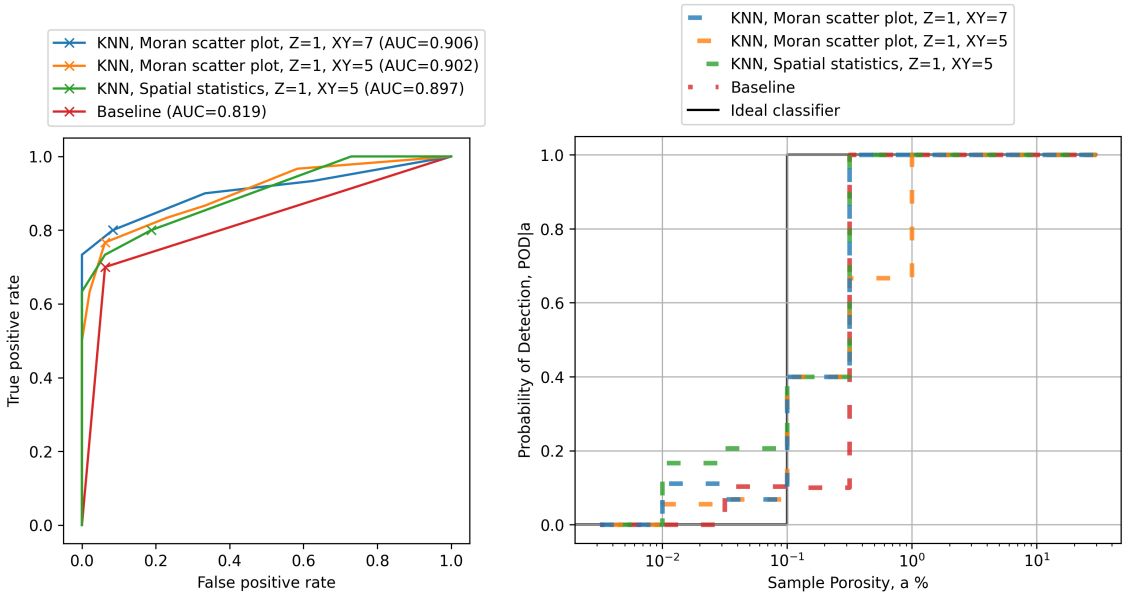
(b) KNN, Moran scatter plot, Z=1, XY=5



(c) KNN, Moran scatter plot, Z=1, XY=3

(d) Baseline

Figure 6.15: Confusion matrices for the three best settings and the baseline classifier for the 0.25% porosity threshold of the V-set evaluation



(a) ROC plot **(b)** POD plot

Figure 6.16: Receiver operating characteristics (ROC) and probability of detection (POD) plots for the three best settings for the 0.10% porosity threshold of the V-set evaluation. The crosses mark the location of the confusion matrices and pod plot

Figure 6.16 shows a ROC curve and POD plot and figure 6.17 shows the confusion matrices for the baseline classifier and the top three settings.

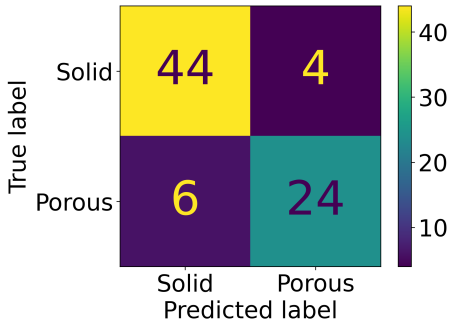
6.3 Discussion of results from the two evaluations

Between the evaluation on the H-set and V-set, a couple of differences as well as similarities were observed. First, the POD plots show that for all thresholds across both evaluations, most methods (including the baseline) are effective at classifying the instances with the highest porosity. This is important, since it is a lot more OK to misclassify an object with 1% porosity as solid compared to an object with 10% porosity. However, focusing on a single metric such as ROC-AUC or F1-score misses that.

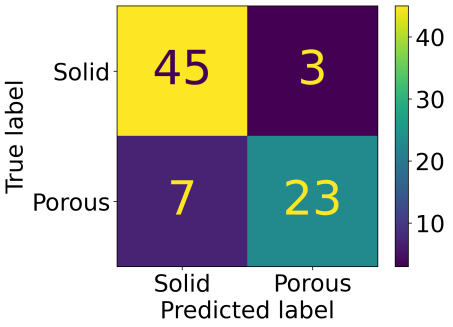
Second, the ideal method/classifier very much depends on the acceptable false positive rate and true positive rate. Across most thresholds in both evaluations, there was no method that clearly was superior.

Third, surprisingly the k-nearest neighbour classifier combined with the Moran scatter plot or spatial statistics performed better in the evaluation on the V-set than the H-set for the 0.25% and 0.5% thresholds, despite not being trained on the V-set. The small size of the data set may contribute to the result being somewhat inflated due to randomness. Furthermore, because of the H-set geometry being slightly more complex, with five segments instead of three. In addition, splitting the object into three parts during the H-set evaluation may also have played a role.

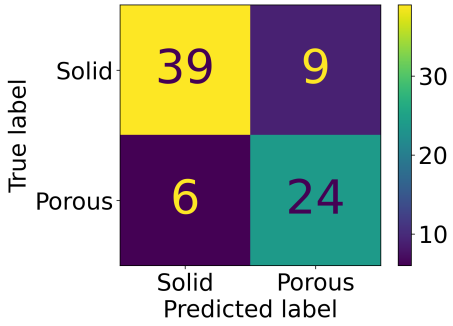
Forth, the spatial statistics and Moran scatter plot tended to generalize well between



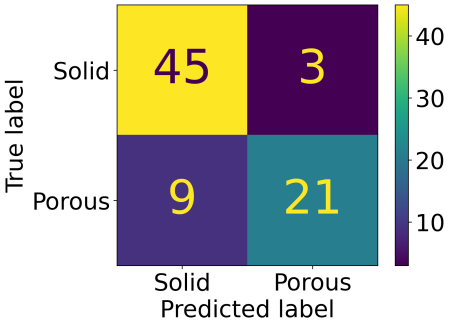
(a) KNN, Moran scatter plot, Z=1, XY=7



(b) KNN, Moran scatter plot, Z=1, XY=5



(c) KNN, Spatial statistics, Z=1, XY=5



(d) Baseline

Figure 6.17: Confusion matrices for the three best settings and the baseline classifier for the 0.10% porosity threshold of the V-set evaluation

the geometries unlike the baseline classifier and the scatter plot. This may be due to both Moran scatter plot and spatial statistics normalizing the data before calculating the outlier values, fitting them in a similar range regardless of the raw grey values, whereas the scatter plot is more effected by the absolute values.
Furthermore, I find it surprising that additional layers (i.e. increasing the Z-length) is not helpful, since this would allow for ignoring areas that are consistently warm between layers, such as edges. It should be noted however that in a study by Feng et al. using a similar set up, the additional layers were indeed helpful [5].
Finally, in the H-set evaluation the baseline classifier was never completely outclassed by another setting, but always offered a TPR/FPR balance that could be considered. This is in stark contrast to the V-set evaluation, where there always was a better classifier.

6.4 Execution Speed

In addition to the evaluation of the classification accuracy, it is of interest to have the execution of the classification be done quickly. In order for it to not slow down the manufacturing process, it should ideally be able to run within the scope of the couple of seconds when the L-PBF machine is conducting recoating between each layer. Figure 6.18 shows a box plot of the execution time. The box extends from the 25th to the 75th percentile, with the median being marked in red. The whiskers show the smallest and largest values. The values were obtained by running each of the methods presented in the evaluation (apart from the baselines) 20 times each, resulting in 20*3*3*2 = 360 data points. The time was measured using the *time* module in Python, and the computation was run on a MacBook Pro 2022 with 16 GB of memory.

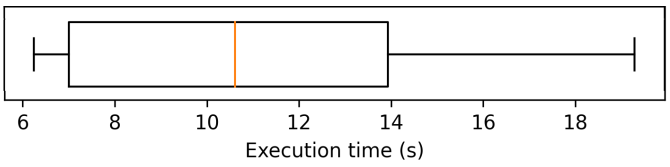


Figure 6.18: Box plot of the execution time

As can be seen in the plot, the classification takes significantly longer than the available time between each layer. However, since the classification reads and processes all data at the same time, there is ample room for improvement. The execution time could likely be reduced significantly by treating the data as a stream and reading and processing the data as it becomes available, one layer at a time.

6.5 Comparison to Previous Research

As previously discussed in chapter 4, there are a number of studies that have used similar methods to detect porosity. Table 6.3 shows an overview of the results of this thesis compared with the state of the art.

The closest comparison in terms of methodology is to de Winton et al., who used the same ground truth data (i.e. optical microscopy images) but different features (co-axical camera readings). The result of the thesis are in line with theirs, except for at the 0.1% threshold where their method performed better. They also included POD plots (although only for the 0.5% threshold), which are similar to the ones in this thesis.
In contrast, Estalaki et al. [13] used similar features (off-axical camera readings) but different ground truth data (X-ray CT images). It is hard to compare their results to the ones in this thesis for two reasons. First, they considered a different problem, namely predicting the porosity of individual voxels of an object. Predicting whether an individual voxel is porous or not is likely to be more useful than predicting the porosity of an entire object, since it allows accounting for a single large porous area being worse than a lot of small ones. In addition, in the interest of being able to fix defects (something that needs additional research) it is important to know where defects are located. Second, although they have precision recall curves (which are equivalent to ROC curves) and although they show bar charts of the ROC-AUC, they stop short of actually stating what the score was. They do however state that their best performance was reached using a random forest classifier, for which their best F1-score was 0.966.
When it comes to the two alternative evaluations conducted in this thesis (namely investigating the performance of generalization between geometries and computational speed), this thesis is to the best of my knowledge the first research to investigate that. Here, the thesis serves as a first step, and the results indicate that it may be feasible to run the classification in real time if the data is treated as a stream instead of as a chunk. Furthermore, the results indicate that there is the possibility to generalize between geometries.

Table 6.3: Comparison of methods investigated with existing methods

Method/Paper	Porosity Threshold	ROC-AUC (or other metric)	Execution Time
This thesis	0.5%	0.989 / 0.999	10.9s
This thesis	0.25%	0.975 / 0.989	10.8s
This thesis	0.1%	0.940 / 0.906	11.2s
de Winton et al. [3]	0.5%	0.98	-
de Winton et al. [3]	0.25%	0.95	-
de Winton et al. [3]	0.1%	0.98	-
Estalaki et al. [13]	0.5%[1]	0.966 F1-Score	-
Feng et al. [5]	-[2]	>0.95 Pearson's r	-

[1]Considered voxels of size 130 x 135 x 50µm^3, not entire objects
[2]Modelled it as a regression problem, i.e. predicting the level of porosity

7

Conclusion and Future Work

This chapter provides a conclusion to the findings of this thesis and how it relates to the existing research, as well as goes over some future areas of research that could be explored.

7.1 Conclusion

In this thesis, it was found that spatial outlier detection methods can be used to identify porous objects in laser powder bed fusion (L-PBF). Furthermore, it was found that there is the possibility to generalize between similar geometries. However, making comparisons to existing research is difficult due to a lack of shared data sets and agreed upon benchmarks. In addition, both this thesis and most existing research have only considered fairly simple geometries, such as cubes and cylinders, and the research has been conducted by creating objects with artificial defects by setting some component of the volumetric energy density to an unsuitable value. As such, it remains to be seen how well the methods presented in this thesis as well as the existing research are able to generalize to more complex geometries and defects that occur naturally in the manufacturing process.

7.2 Future Work

Based on the literature survey and results of the thesis, I have identified the following potential areas for future work in the field:

- Investigating how well models trained on one geometry generalize to other geometries. In this thesis, it appears that between the two geometries investigated the models generalize well. However, the two geometries investigated are similar. In particular, it would be interesting to investigate how well models can generalize from geometries such as cylinders and cubes to ones that are more prone to defects, such as overhangs, and geometries that are unique to additive manufacturing, such as lattice structures.
- Relating to generalizing between geometries, it would be interesting to investigate the use of transfer learning to obtain better results. Since producing new data is expensive, one way to overcome this could be to first train a machine learning model on one data set and then on another. This could be used to first train a model on a large, publicly available data set and then retraining it on a smaller data set of a different material, or a different geometry that

is closer to the one of interest. Unfortunately, as mentioned in the previous section, there is a lack of publicly available data sets for the research field.

- Improving upon the methodology presented in this thesis. There are numerous areas that could result in better results, such as using investigating more machine learning models with more hyperparameter settings, more method of spatial outlier detection, more methods of aggregation, upsampling of the data etc. This would be of particular interest if applied in a real-world setting, where the goal is to obtain the best possible results.
- Investigating the feasibility of *federated learning*, i.e. training a model on data that is distributed across multiple devices. This could be used to train a model on data from multiple 3D printers, or data from multiple materials, without the researchers directly sharing information about the geometry constructed, which may be valuable in a production setting.
- Combining multiple existing methods. The research has looked at using one form of data from one sensor. This could be complemented with other types of data (such as using both a co- and off-axical camera) to produce a more accurate model.
- Investigating the possibility to fix defects in previous layers. Estalaki et al. showed that the sensor readings of future layers have predictive power for the current layer [13] and Feng et al. [5] showed that if the laser power is too low in one layer, changing it to a better in the next layers can fix some defects. This could be investigated further, for example by investigating different settings for the laser power.
- Establish common data sets and benchmarks for evaluation. de Winton et al. [3] have proposed a method and metrics for the research to follow, but without a common data set it is difficult to compare results.
- Computational efficiency. As seen in this thesis, despite using a setup producing a low amount of data (i.e. an off-axical camera setup producing a single image per layer), the timing constraints are still not trivial to meet. In order for the methods to be applicable in a production setting, there is a need to make the methods more efficient and benchmark the execution speed of in-situ monitoring methods, in particular when using sensor setups with a higher spatial or temporal resolution than the one used in this thesis.

Bibliography

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, July 2009.

[2] Zhuoer Chen, Xinhua Wu, and Chris H. J. Davies. Process variation in Laser Powder Bed Fusion of Ti-6Al-4V. *Additive Manufacturing*, 41:101987, May 2021. Publisher: Elsevier B.V.

[3] Henry C. de Winton, Frederic Cegla, and Paul A. Hooper. A method for objectively evaluating the defect detection performance of in-situ monitoring systems. *Additive Manufacturing*, 48:102431, December 2021.

[4] Mohit Dharnidharka, Utkarsh Chadha, Lohitha Manya Dasari, Aarunya Paliwal, Yash Surya, and Senthil Kumaran Selvaraj. Optical tomography in additive manufacturing: a review, processes, open problems, and new opportunities. *The European Physical Journal Plus*, 136(11):1133, November 2021.

[5] Shuo Feng, Zhuoer Chen, Benjamin Bircher, Ze Ji, Lars Nyborg, and Samuel Bigot. Predicting laser powder bed fusion defects through in-process monitoring data and machine learning. *Materials & Design*, 222:111115, October 2022.

[6] B. K. Foster, E. W. Reutzel, A. R. Nassar, B. T. Hall, S. W. Brown, and C. J. Dickman. Optical, Layerwise Monitoring of Powder Bed Fusion. University of Texas at Austin, 2015. Accepted: 2021-10-19T19:20:04Z.

[7] Zhang Fu, Magnus Almgren, Olaf Landsiedel, and Marina Papatriantafilou. Online temporal-spatial analysis for detection of critical events in Cyber-Physical Systems. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 129–134, Washington, DC, USA, October 2014. IEEE.

[8] Christian Gobert, Edward W. Reutzel, Jan Petrich, Abdalla R. Nassar, and Shashi Phoha. Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Additive Manufacturing*, 21:517–528, May 2018.

[9] Marco Grasso and Bianca Maria Colosimo. Process defects and *in situ* monitoring methods in metal powder bed fusion: a review. *Measurement Science and Technology*, 28(4):044005, April 2017.

[10] Ohyung Kwon, Hyung Giun Kim, Min Ji Ham, Wonrae Kim, Gun-Hee Kim, Jae-Hyung Cho, Nam Il Kim, and Kangil Kim. A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing*, 31(2):375–386, February 2020.

[11] Cody S. Lough, Tao Liu, Xin Wang, Ben Brown, Robert G. Landers, Douglas A. Bristow, James A. Drallmeier, and Edward C. Kinzel. Local prediction of Laser Powder Bed Fusion porosity by short-wave infrared imaging thermal

feature porosity probability maps. *Journal of Materials Processing Technology*, 302:117473, April 2022.

[12] Oded Maimon and Lior Rokach, editors. *Data Mining and Knowledge Discovery Handbook*. Springer US, Boston, MA, 2010.

[13] Sina Malakpour Estalaki, Cody S. Lough, Robert G. Landers, Edward C. Kinzel, and Tengfei Luo. Predicting Defects in Laser Powder Bed Fusion Using In-Situ Thermal Imaging Data and Machine Learning. *SSRN Electronic Journal*, 2022.

[14] Ronan McCann, Muhannad A. Obeidi, Cian Hughes, Éanna McCarthy, Darragh S. Egan, Rajani K. Vijayaraghavan, Ajey M. Joshi, Victor Acinas Garzon, Denis P. Dowling, Patrick J. McNally, and Dermot Brabazon. In-situ sensing, process monitoring and machine control in Laser Powder Bed Fusion: A review. *Additive Manufacturing*, 45:102058, September 2021.

[15] Gunther Mohr, Simon J. Altenburg, Alexander Ulbricht, Philipp Heinrich, Daniel Baum, Christiane Maierhofer, and Kai Hilgenberg. In-Situ Defect Detection in Laser Powder Bed Fusion by Using Thermography and Optical Tomography—Comparison to Computed Tomography. *Metals*, 10(1):103, January 2020.

[16] Mohammad Montazeri, Abdalla R. Nassar, Alexander J. Dunbar, and Prahalada Rao. In-process monitoring of porosity in additive manufacturing using optical emission spectroscopy. *IISE Transactions*, 52(5):500–515, May 2020.

[17] Kevin P. Murphy. *Machine Learning : A Probabilistic Perspective*. MIT Press, Cambridge, UNITED STATES, 2012.

[18] Kwok-suen Ng, Wing-tat Hung, and Wing-gun Wong. An algorithm for assessing the risk of traffic accident. *Journal of Safety Research*, 33(3):387–410, October 2002.

[19] Negar Panahi. *Multi-purpose parameter development for high productivity in Laser Powder Bed Fusion of IN718*. PhD thesis, Chalmers University of Technology, Gothenburg, 2020.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Claudia Schwerz and Lars Nyborg. A neural network for identification and classification of systematic internal flaws in laser powder bed fusion. *CIRP Journal of Manufacturing Science and Technology*, 37:312–318, May 2022.

[22] Robert Snell, Sam Tammas-Williams, Lova Chechik, Alistair Lyle, Everth Hernández-Nava, Charlotte Boig, George Panoutsos, and Iain Todd. Methods for Rapid Pore Classification in Metal Additive Manufacturing. *JOM*, 72(1):101–109, January 2020.

[23] Andreas Wegner and Gerd Witt. Process monitoring in laser sintering using thermal imaging. *22nd Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2011*, January 2011.

[24] Guenter Zenzinger, Joachim Bamberg, Alexander Ladewig, Thomas Hess, Benjamin Henkel, and Wilhelm Satzger. Process monitoring of additive manufacturing by using optical tomography. pages 164–170, Boise, Idaho, 2015.

[25] Bin Zhang, Shunyu Liu, and Yung C. Shin. In-Process monitoring of porosity during laser additive manufacturing process. *Additive Manufacturing*, 28:497–505, August 2019.

A

Appendix 1

This appendix contains additional data from the evaluation. The tabels are color coded, with green signifying a higher score than the baseline classifier, red a lower score and white a score that was similar to the baseline.

Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	1	5	10	0.951
	scatter plot	1	7	40	0.978
	spatial statistics	1	3	40	0.951
KNN	Moran scatter plot	1	7	40	0.987
	scatter plot	1	3	20	0.989
	spatial statistics	1	7	40	0.967
Baseline classifier		-	-	-	0.981

(a) 0.50% Threshold

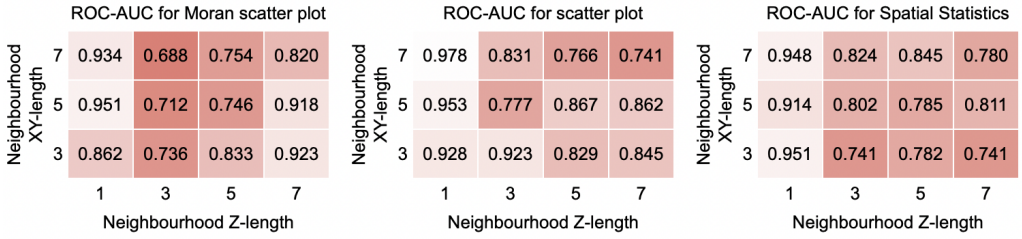
Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	1	5	10	0.969
	scatter plot	1	7	40	0.946
	spatial statistics	1	3	40	0.939
KNN	Moran scatter plot	1	5	40	0.962
	scatter plot	3	5	40	0.975
	spatial statistics	1	7	40	0.956
Baseline classifier		-	-	-	0.950

(b) 0.25% Threshold

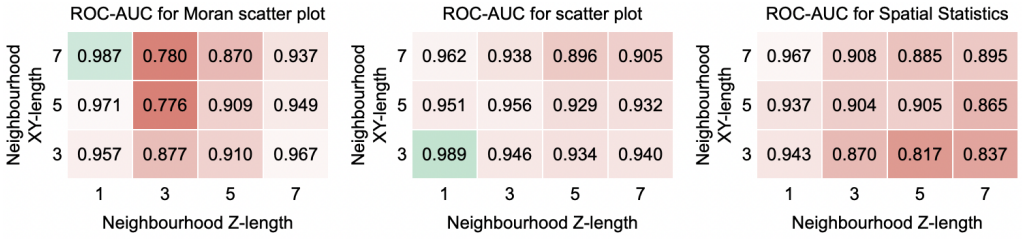
Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	7	5	40	0.842
	scatter plot	1	5	20	0.846
	spatial statistics	1	7	20	0.810
KNN	Moran scatter plot	1	7	20	0.940
	scatter plot	1	5	40	0.923
	spatial statistics	1	7	10	0.887
Baseline classifier		-	-	-	0.880

(c) 0.10% Threshold

Table A.1: ROC-AUC for the best combination of Z-length and XY-length for each classifier and outlier detection method combination in the H-set evaluation. Green indicates better results than the baseline classifier, red indicates worse results than the baseline classifier

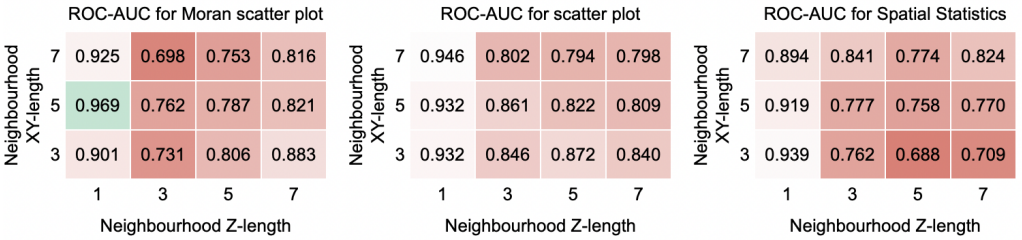


(a) Decision tree

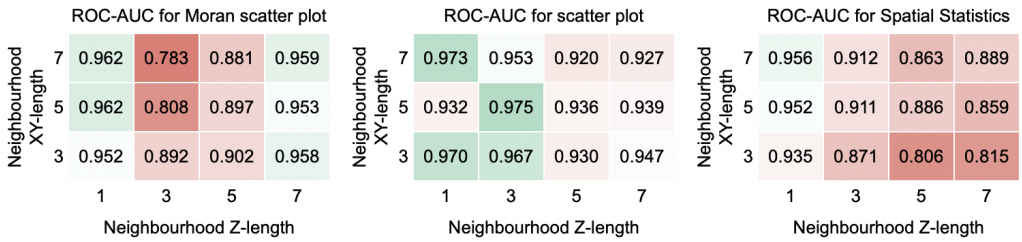


(b) K-nearest neighbour

Table A.2: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.50% porosity during the evaluation on the H-set. The baseline classifier had a ROC-AUC of 0.93: values at this level are white, with worse values being progressively darker red and better values progressively darker green.

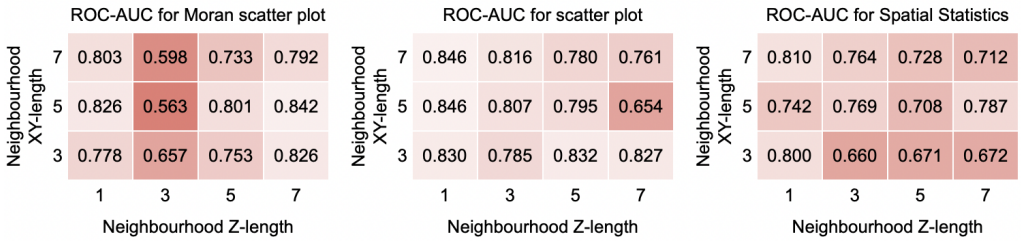


(a) Decision tree

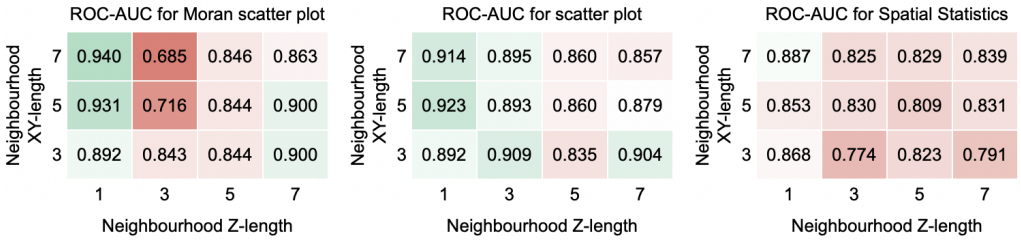


(b) K-nearest neighbour

Table A.3: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.25% porosity during the evaluation on the H-set. The baseline classifier had a ROC-AUC of 0.89: values at this level are white, with worse values being progressively darker red and better values progressively darker green.



(a) Decision Tree



(b) K-nearest neighbour

Table A.4: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.10% porosity during the evaluation on the H-set. The baseline classifier had a ROC-AUC of 0.81: values at this level are white, with worse values being progressively darker red and better values progressively darker green.

Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	7	7	20	0.867
	scatter plot	1	5	20	0.808
	spatial statistics	1	7	40	0.933
KNN	Moran scatter plot	1	7	40	0.994
	scatter plot	3	5	40	0.859
	spatial statistics	1	7	40	0.999
Baseline classifier		-	-	-	0.973

(a) 0.50% Threshold

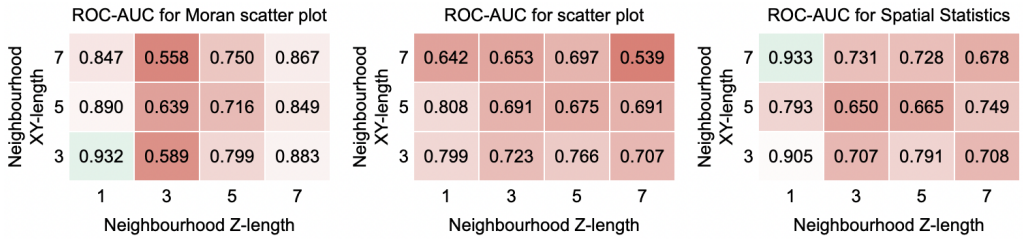
Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	1	3	40	0.932
	scatter plot	1	5	10	0.808
	spatial statistics	1	3	40	0.905
KNN	Moran scatter plot	1	7	40	0.994
	scatter plot	3	5	40	0.858
	spatial statistics	1	3	10	0.975
Baseline classifier		-	-	-	0.875

(b) 0.25% Threshold

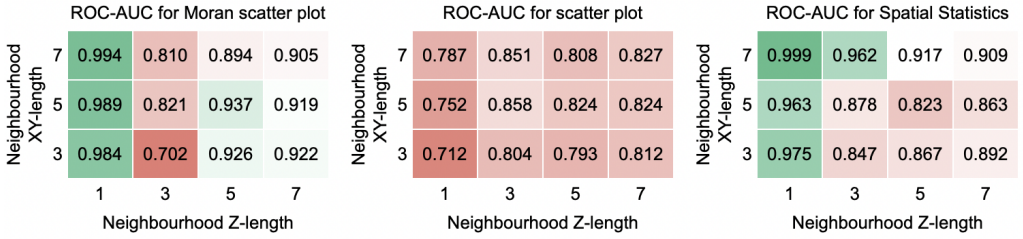
Classifier	Outlier detection method	Z-length	XY-length	Bins	Test ROC-AUC
DT	Moran scatter plot	1	5	40	0.808
	scatter plot	1	7	10	0.729
	spatial statistics	3	3	20	0.731
KNN	Moran scatter plot	1	7	20	0.906
	scatter plot	7	7	10	0.800
	spatial statistics	1	5	5	0.897
Baseline classifier		-	-	-	0.819

(c) 0.10% Threshold

Table A.5: ROC-AUC for the best combination of Z-length and XY-length for each combination of classifier and outlier detection method in the V-set evaluation. Green indicates better results than the baseline classifier, red indicates worse results than the baseline classifier

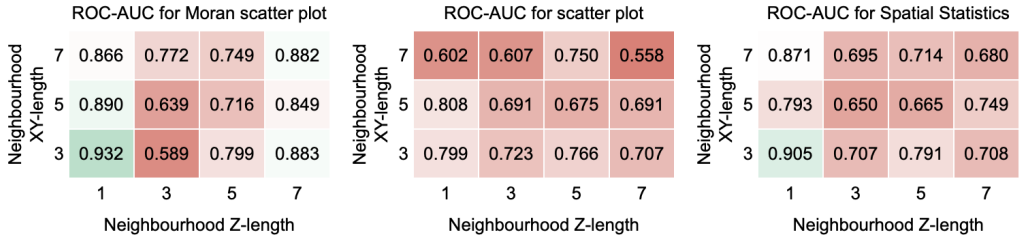


(a) Decision tree

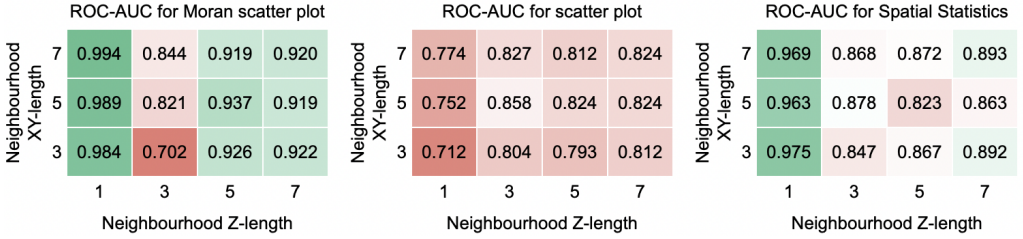


(b) K-nearest neighbour

Table A.6: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.50% porosity during the evaluation on the V-set

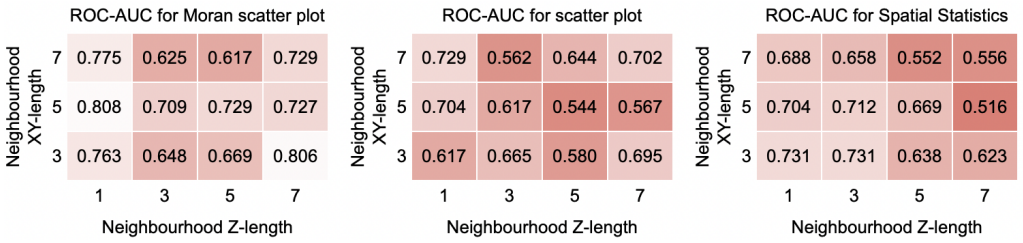


(a) Decision tree

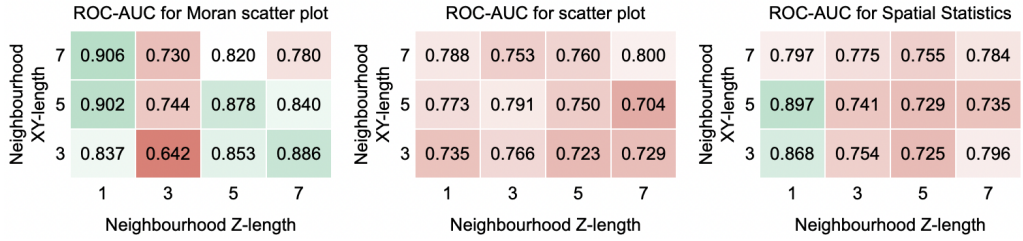


(b) K-nearest neighbour

Table A.7: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.25% porosity during the evaluation on the V-set



(a) Decision Tree



(b) K-nearest neighbour

Table A.8: ROC-AUC for the two classifiers for different outlier detection methods and different neighbourhood sizes at 0.10% porosity during the evaluation on the V-set