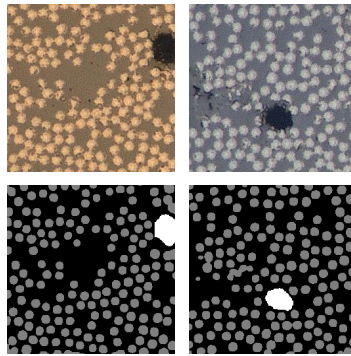


UNIVERSITY OF TWENTE.



A Deep Learning Approach for Assessing Micrographs of Fibre Reinforced Composite Laminates



M. M. Fernhout
Master Thesis
October 2023

Exam committee

Prof.dr.ir. C.H. Venner
Dr.ir. M.I. Abdul Rasheed
Dr.ir. W.J.B. Groeve
Dr.ir. J.A.M. Withag

Report number

437

**Engineering Fluid Dynamics
Faculty of Engineering Technology
Mechanical Engineering
University of Twente
The Netherlands**



Preface

Dear reader,

In front of you lies my master thesis titled: “A Deep Learning Approach for Assessing Micrographs of Fibre Reinforced Composite Laminates”. This thesis marks the end of my master in Mechanical Engineering with a specialisation in Energy & Flow at the University of Twente.

During the final courses of my master I encountered machine learning. I was captivated and wanted to explore this subject further, so I searched for a master’s thesis assignment related to this. I am pleased to have been introduced to this assignment that combines machine learning with a familiar acquaintance: fibre reinforced polymers. These materials first crossed my path during my gap year at Green Team Twente, where I built a hydrogen car from carbon fibre reinforced polymers with an amazing group of students. Furthermore, my bachelor’s thesis delved into the effects of heat treatment on the reconsolidation repair of a PAEK thermoplastic composite. Therefore, when I came across this research project, it seemed like a perfect match, and I am still happy with the decision I made.

This thesis is the result of a collaborative effort involving three research groups at the university: Engineering Fluid Dynamics, Applied Mechanics & Data Analysis and Production Technology. I am grateful to Kees Venner for making this possible and allowing me to choose a topic that diverged a bit from my Energy & Flow specialisation. Your valuable insights during our meetings were greatly appreciated. Most importantly, I want to thank my supervisors, Iqbal Rasheed and Wouter Grouve, because this work would not be possible without your support. I cannot thank you enough for always being available for questions and the feedback given every week was greatly appreciated.

Furthermore, I would like to thank some people at TPRC for providing me with their materials and/or micrographs to create the data sets. My sincere thanks go to Rens Pierik, Yannick Buser, Tom Asijee, Luis Gonzalez Camacho and Thijs Rouwmaat. It was not an easy task to collect enough images to train the models and their assistance was indispensable in achieving this.

Lastly, I would like to thank my friends and family for their endless support and motivation. You kept me motivated throughout this journey, whether it was the walks during lunch break that brightened my days or the shared dinners and comforting phone calls that kept me going.

Thank you for taking the time to read my thesis.

Merit Fernhout
Enschede, October 2023

Abstract

Fibre reinforced composites are becoming more common in a world that demands lightweight products. These materials play a crucial role in the aerospace and automotive industries in reducing fuel consumption. Additionally, they are essential in sustainable energy technologies, particularly for the production of improved wind turbines. Fibre reinforced composites consist of high-performance fibres, such as carbon or glass, embedded in a matrix material. The distribution of fibres in the matrix material and the presence of defects, such as air inclusions, greatly influence the mechanical performance of the material. Microscopic images of cross-sections of the material can be used to visualise these characteristics so that a human can identify any possible defects. However, automating this analysis would be highly beneficial and time-efficient. Currently, image analysis techniques are mainly based on manual pixel intensity thresholding, which is sensitive to illumination conditions during image generation and image quality. Machine learning offers a potential solution to automate the analysis of the micrographs. Therefore, the objective of this research is to explore the potential of using a machine learning model to recognise fibres and voids in micrographs of composites to determine the void and fibre volume fractions in the material. This thesis proposes a deep learning approach with a model based on a u-net architecture. The u-net model is trained, validated and tested with data sets composed of images of 256 x 256 pixels cut from microscopy images of carbon fibre composites. This research also includes the generation of these data sets with corresponding ground truth masks of the images. Furthermore, data augmentation is used to increase diversity in these training data sets, which was found to improve the prediction results of the trained models on the test data. The trained models demonstrate that this deep learning approach is capable of accurately recognising voids and fibres without the need for calibration, unlike traditional thresholding techniques. Thus, it is shown that this deep learning approach is a promising method for identifying fibres and voids in microscopy images, despite having a limited data set and not having ideal ground truth masks.

Contents

1	Introduction	5
1.1	Problem definition	6
1.2	Research goal	7
1.3	Outline	7
2	Background	9
2.1	Semantic segmentation	9
2.2	Machine learning	9
2.3	Deep learning	10
2.3.1	Neural networks	11
2.4	Overfitting	12
2.5	Regularisation	12
2.5.1	L2 regularisation	13
2.5.2	Dropout	13
2.6	Convolutional neural networks	13
2.6.1	Convolutional layers	14
2.6.2	Activation functions	15
2.6.3	Pooling layers	16
2.6.4	Batch normalisation layers	17
2.7	U-net	17
3	Data	20
3.1	Micrographs	20
3.2	Image sets	21
3.3	Mask generation	23
3.3.1	Fibres	24
3.3.2	Voids	27
3.4	Data sets	32
3.4.1	Data set I	32
3.4.2	Data set II	33
3.4.3	Data set III	34
3.4.4	Test data set	35
3.5	Data augmentation	36
4	U-net model	39
4.1	U-net architecture	39
4.2	Training, validating and testing	40
4.3	Categorical cross-entropy	41
4.4	Evaluation metrics	41
4.4.1	Accuracy	41
4.4.2	Intersection over Union	43

5	Results	44
5.1	Model I: Augmented data set I	45
5.2	Model II: Augmented data set II	47
5.3	Model III: Augmented data set III	49
5.4	Comparison of the three models	52
6	Discussion	53
6.1	Mask generation	53
6.1.1	Fibre and void volume fraction	55
6.2	Effect of various parameters on model III	55
6.2.1	Data augmentation	55
6.2.2	Amount of data	56
6.2.3	Model hyperparameters	57
7	Conclusion	60
8	Recommendations	61
9	References	63
A	Micrograph details	68
A.1	Source 1	69
A.2	Source 2	70
A.3	Source 3	70
A.4	Source 4	70
A.5	Source 5	70
B	Data sets	71
C	Results of models trained with non-augmented data sets	72
C.1	Data set I	72
C.2	Data set II	74
C.3	Data set III	76
D	Comparison of model III and the original u-net model	79
E	One fibre prediction	80
F	Feature maps final model	80

1 Introduction

Fibre reinforced polymer (FRP) composites are becoming more common in a world that demands lightweight products. These materials play a crucial role in the aerospace and automotive industries by reducing fuel consumption. Additionally, they are essential in sustainable energy technologies, particularly for the production of state-of-the-art wind turbines [1], as all wind turbines today are made up of blades made up of FRP composites [2].

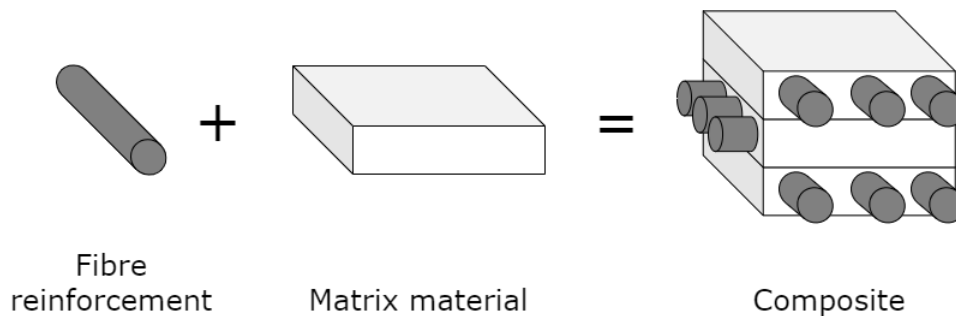


Figure 1: A composite material is built up of fibres and matrix material. These can be stacked into different layers to get a composite laminate.

FRPs are composite materials that have three major constituents: high-performance fibres, matrix material, and voids [3]. Fibres mainly carry the load of the material and provide strength and stiffness to it. Carbon and glass fibres are commonly used for this purpose. The matrix material binds these fibres together and transfers loads between them, as can be seen in Figure 1. The matrix material can be either a thermoset resin or a thermoplastic polymer. In addition to fibres and matrix material, composites may contain voids. Voids are gas bubbles trapped in the matrix material and can arise during the manufacturing of the composite. They are highly unwanted, and even at a low volume percent, they can significantly affect the properties of the FRP. A higher void content results in a decrease in mechanical properties. Aerospace-grade composites often exhibit void contents of around 1%, while other composites can have void contents between 3 and 5%. An increase in this void volume fraction from only 1 to 3% in highly loaded FRPs can result in a 20% loss in the mechanical properties of the laminate [4]. The void volume fraction V_v can be calculated as:

$$V_v = \frac{V_{\text{voids}}}{V_{\text{composite}}}, \quad (1)$$

where V_{voids} is the total volume of the voids inside the composite and $V_{\text{composite}}$ is the total volume of the whole composite.

To understand the behaviour of materials, it is crucial to understand and know the properties of materials. In addition to analysing the void content, characterising the microstructure of materials can help to understand the properties of the material. This is especially important in the case of (carbon) composites, since the matrix material and the reinforcement material can be arranged in many possible ways [5]. Therefore, the arrangement of the fibres in the composite, like orientation and geometry, is one of the factors that typically determines the

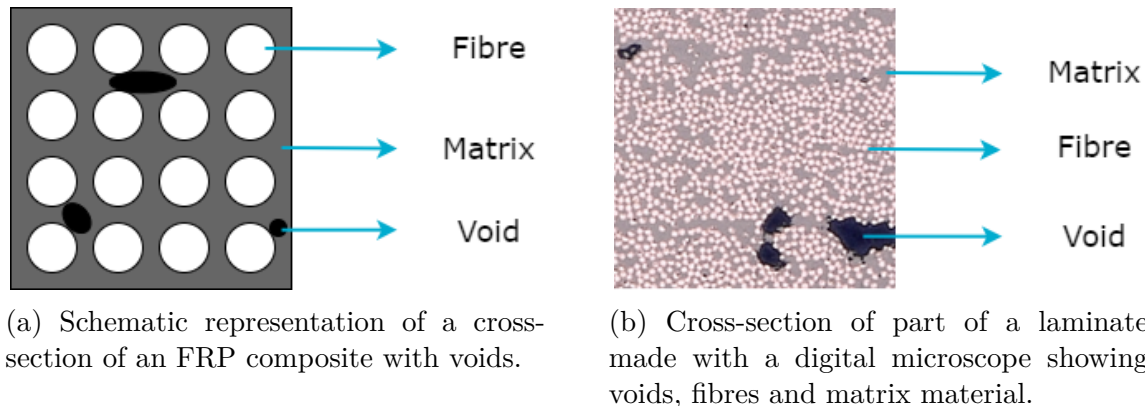


Figure 2: Visual representation of voids in cross-sections of FRP material.

properties of a composite. In addition, the fibre properties, matrix properties, the fibre volume fraction and the void volume fraction [6] also determine the properties of the composite. Similarly to the volume fraction of the void, the volume fraction of the fibres V_f of an FRP composite can be calculated as:

$$V_f = \frac{V_{\text{fibre}}}{V_{\text{composite}}}. \quad (2)$$

Here, V_{fibre} is the volume of all fibres in the composite and $V_{\text{composite}}$ is the total volume of the composite. Therefore, V_f is the percentage of the volume of the composite made up of fibres. This ratio is typically in the range of 50-70% [7]. The V_f of an FRP composite is highly dependent on the stacking of the fibres and the manufacturing process, especially the pressure applied during the curing processes. The strength of the composite tends to increase with the amount of fibre it contains since the strength of the fibres is usually greater than that of the matrix material. However, studies show that after a certain V_f the strength can decrease again because there is not enough matrix material to wet the fibres. This leads to debonding of the composite and premature failure. In the paper ‘‘Effect of fiber-matrix volume fraction and fiber orientation on the design of composite suspension system’’ Ali and Anjaneyulu [8] demonstrated that this strength decreases after a V_f of 70%.

1.1 Problem definition

The fibre volume fraction is generally determined by destructive methods. According to ASTM D3171-22 [9], the industry standard for determining the volume fraction of fibre, the matrix is removed by digestion, ignition, or carbonisation. This leaves the fibres essentially unaffected and can thus be used to calculate the content of the fibres and the matrix. This method can even be used to determine the void volume fraction by comparing the composite density to the acquired matrix and fibre volume fractions. In addition to density-based methods, microCT, ultrasonic testing, thermography, and optical or electron microscopy are also commonly used [10].

Optical microscopy is the most widely used method to evaluate the content of the void. It is

relatively simple and inexpensive and provides more detail than ultrasound and thermography. Especially when spatial distributions of the fibres are required or characteristics of voids, like their dimensions, shape, and number count. However, optical microscopy is a destructive method. The laminate is cut to obtain a cross-section. This cross-sectional area can be analysed with an optical microscope. Analysing the fibre and void content in these micrographs manually is a time-consuming process and, therefore, not preferred.

A commonly used analysis method is image segmentation using pixel intensity thresholding [11]. This method uses the intensities of the pixels of fibres, voids, and matrix material in the images. Thresholding values are determined for every image to identify pixels that belong to fibres, voids, and the matrix. However, these thresholding values need to be calibrated for every image due to variations in image quality and illumination conditions. This technique is not effective in dealing with the differences observed in microscopy images, whereas humans can identify fibres, gaps, and matrix material with ease, regardless of the image's quality.

Machine learning algorithms have already been shown to be capable of automatically detecting voids in micrographs. Luo et al. [12] developed a void recognition model based on a DeepLabV3+ model. This model could effectively and automatically detect void characteristics in microscopy images. Furthermore, Machado et al. [11] successfully demonstrated that the voids in micrography samples could be automatically segmented using a machine learning model based on a u-net architecture. These results exceeded the performance of a thresholding-based algorithm, which requires manual calibration and can be used to determine the relative void content. However, no examples have been found that could identify voids and fibres to analyse both contents. Therefore, this study will investigate the possibilities of using a single machine learning model to identify fibres and voids to calculate the fibre and void content.

1.2 Research goal

The goal of this thesis is to investigate the possibilities of using machine learning techniques to analyse microscopy images of cross-sections from FRP laminates to determine the volume fractions of the void and the fibres. Thereby, implementing, training, and validating machine learning algorithms to explore the potential of creating a more automated image analysis method that can overcome the thresholding problem of having variations in image quality. This research will involve the development of a data set based on actual microscopy images that will be used to train and evaluate machine learning algorithms.

1.3 Outline

The following chapters of this thesis are organised as follows. Chapter 2 provides the essential background information needed to understand the rest of the thesis. Machine learning and deep learning will be explained here, as well as the segmentation problem and the u-net architecture. The methodology of this research is split into two parts: data and the u-net model. Chapter 3 provides an overview of the images used to create four different data sets and how their corresponding ground-truth masks are generated. Chapter 4 explains the

specific architecture of the u-net model and the parameters used during training, along with the evaluation metrics used to assess the performance of the model. In Chapter 5 the results of the three trained models are presented and compared. The best-performing model is then studied to discuss the effect of changing certain parameters and the input training data in Chapter 6. In this chapter, the results of the mask generation process will also be discussed. Finally, the conclusions of this study are presented in Chapter 7 and recommendations for further research are discussed in Chapter 8.

2 Background

This chapter provides a brief overview of the background knowledge necessary to comprehend the thesis. Semantic segmentation will be discussed first, followed by the machine learning concepts used in this study. Those with prior experience in machine learning may be able to skip this chapter.

2.1 Semantic segmentation

The objective of this research is to analyse cross-sectional microscopy images of FRP material and to determine which part of the image shows fibres, matrix material, and voids. This goes beyond simply classifying which animal (e.g. dog, cat, or elephant) is visible in an image, since it also focuses on the area of the image that contains the object, as the location of the fibres is also important. In Figure 3 an image of two elephants is shown to compare different computer vision technologies. Figure 3a shows the previously discussed example of a classification problem. Object recognition attempts to identify one or more objects within an image or video by predicting a rectangular bounding box around the object of interest, as can be seen in Figure 3b. Image segmentation goes a step further by partitioning an image into segments, a group of pixels, that more precisely define an object. It assigns the same labels to pixels in images that share specific features. Semantic segmentation detects for every pixel to which class it belongs. So in Figure 3c all pixels belonging to elephants are detected. While instance segmentation also identifies each specific instance of an object; the two elephants are segmented individually as can be seen in Figure 3d [13]. In this thesis, pixels belonging to fibres, matrix material and voids are of interest, but not the fibres and voids individually, as this would make the model overly complicated, which makes this a semantic segmentation problem.

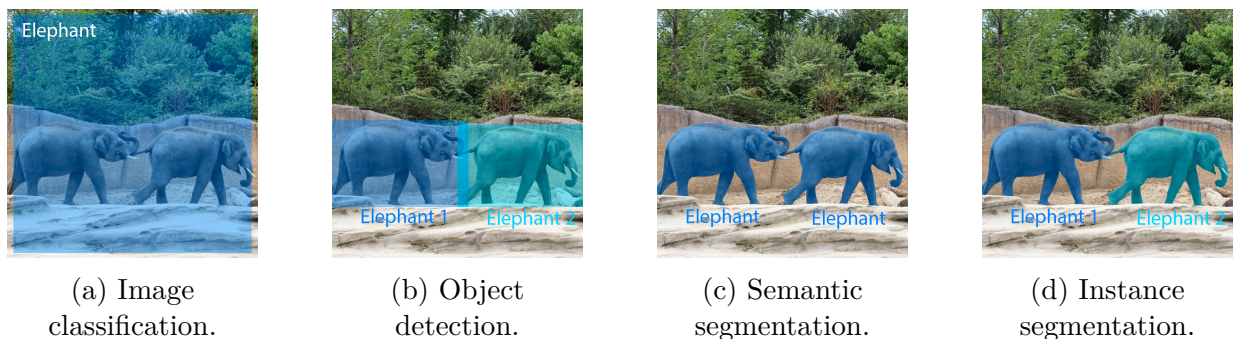


Figure 3: Different computer vision techniques to detect elephants in an image.

2.2 Machine learning

The use of machine learning, particularly deep learning, is becoming increasingly common in the field of image processing for image segmentation tasks [14]. Machine learning is the technique that requires a system to learn from experience via computational methods rather than being explicitly told what to do by a human-developed algorithm. Experience in computer systems usually comes in the form of data. By feeding data to a learning algorithm,

a model can be obtained that can make predictions on newly observed data. The process of machine learning algorithms to construct models from training data is known as learning or training [15]. Machine learning can be roughly divided into three main types: supervised learning, unsupervised learning, and reinforcement learning. With supervised learning, a model is given both input and corresponding output during training. For example, a model is fed images of dogs and cats and is told the output (label) of each image. The goal is to learn a general principle that maps the inputs to the outputs, in a way that once the model has been trained, it can accurately predict the class (e.g. cat or dogs) of unlabelled images. In the case of unsupervised learning, no labels are given during training. The learning algorithm needs to identify patterns and structures in the data. It can, for instance, cluster the pictures of cats and dogs into two distinct classes by itself, but it does not have the information that these classes contain cats and dogs. Lastly, reinforcement learning is based on learning from interaction with an environment and rewarding desired behaviour and/or punishing unwanted behaviour [16]. In this research, the focus is on supervised learning, for which three main components are required to build a successful model:

- Input images,
- Expected outputs corresponding to the input images,
- Performance metrics.

First, the input data will be microscopy images of cross-sections of composite materials. Second, examples of expected outputs are needed. Thus, a ground truth image needs to be created of the input images that acts as a mask and tells the algorithm to which class each of the pixels in the input image belongs. Third, performance measures are needed to connect the input to the output and provide feedback to correct the algorithm so it better matches the expected output.

2.3 Deep learning

Deep learning is a branch of machine learning that uses algorithms inspired by the human brain to address a wide range of machine intelligence tasks. Many examples can be seen in everyday life. From face detection on your phone to auto-correction on keyboards and autopilots in cars [13, 17]. These algorithms are largely based on artificial neural networks, which use mathematical functions to map a set of inputs to an output [18]. They are composed of a large number of computing cells, known as “neurons”. Each neuron performs simple operations and can interact with other neurons to make decisions [14]. Deep neural networks are neural networks that have multiple layers between the input and output layers. These complex networks have a large number of layers and learn directly from raw data; hence the term “deep”. This is the main distinction between machine learning and deep learning; with deep learning, the machine develops relevant features for the tasks in an automated way instead of having to indicate these features manually, just like humans. When a child receives illustrations of an elephant, they will quickly learn to recognise the animal, without an adult identifying its specific features, such as the trunk, big ears, and tusks. Thus, a human can learn from raw data and draw conclusions without explicitly identifying features. In deep learning, the model is also provided with raw data and can identify useful features, such as corners, edges, and ridges in images, without the need for an engineer to manually

input them. Deep learning has a downside in that it is very computationally intensive and requires a great deal of memory and computational resources. Furthermore, to create a deep learning model that is not prone to overfitting, which will be explained in Chapter 2.4, a large number of labelled images are required and generally result in a long training time [19].

2.3.1 Neural networks

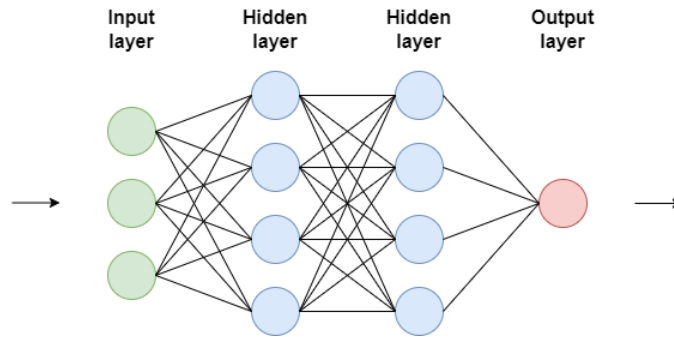


Figure 4: Example of a simple artificial neural network with an input layer, two hidden layers and a final output layer.

An artificial neural network contains an input layer, several hidden layers, and an output layer, as shown in the example in Figure 4. The data is first received by the input layer, which then passes it through the hidden layers. These layers transform the data into useful information for the output layer. To teach a neural network how to produce the desired results, it must be trained. Generally, the layers of neural networks contain values called weights. During training, the weights are updated to optimise the performance of the network. The goal of training a network is to find the weights values that lead to the best results using backpropagation.

Backpropagation is a fundamental building block of neural networks and is used to effectively train a network by backpropagating errors. Before training, the weights of the neural network are randomly selected, making it unlikely that it will generate meaningful predictions on the inputs. The input data is first fed through the network, and the predictions for the given inputs are calculated, which is called a forward pass. After these predictions are made, a loss function is used to minimise the difference between the predicted outputs and the ground truth outputs and to determine how well the network is performing. The form of loss functions can vary and must be carefully selected for each individual use. The loss function selected for this segmentation problem is shown in Equation 9 and will be discussed in Chapter 4.3. The weights are then adjusted so that the loss function values decrease by calculating the gradient of the loss function for the weights during the backward pass. This is called a backward pass, as it proceeds backwards through the network, moving from the output layer to the input layer. Once all gradients have been calculated in the network, a gradient descent algorithm or one of its variants (e.g. the Adam algorithm) is used to modify the weights so that the loss function is minimised.

This process is repeated for multiple epochs during training. One epoch is when the entire training data set is passed through the network. This is often done in batches, a subset of the data set. Therefore, in this research, batch size refers to the number of images used in each forward and backward pass during training. Using smaller batches for training leads to more frequent updates of the weights; however, it can also cause more noise during the process. On the other hand, larger batches can provide smoother updates, but require more memory. Therefore, the batch size is a hyperparameter, which is a value that is used to control the learning process. Another hyperparameter that must be selected is the learning rate, which determines the size of the step taken when the neural network updates the weights during training. The learning rate has a value between zero and one. Networks with small learning rates generally take more epochs to converge, however, larger learning rates can lead to overshooting the optimal weights.

2.4 Overfitting

In machine learning overfitting is a behaviour that occurs when a model produces accurate predictions for the training data but not for new data. This problem arises when the model has been trained on the data, including the noise, to such an extent that it has not been able to capture the underlying general knowledge. Instead, it has learnt to remember the training examples and fails to generalise and thus performs poorly on unseen data. For example, when an overfitted model should identify images that contain elephants, it may not be able to identify those that show elephants in a desert if it has only been trained with pictures of elephants in forests. This is because it may have learnt to use trees as a feature for classification.

When the training data set is too small, it can lead to overfitting, as the training data does not have enough samples to accurately represent all possible input data values. Overfitting can also be caused by having too much irrelevant information in the training data or by having a model that is too complex. In this case, the model will learn the noise in the training data. Additionally, when the model is trained for an extended period of time on a single data set, it can also result in overfitting, as the model can memorise the samples in the training data, which does not generalise well on unseen data.

2.5 Regularisation

Regularisation techniques are frequently used to prevent overfitting and improve the generalisation performance of a model. Regularisation methods used during training add a penalty term to the loss function, which helps to prevent the model from becoming too complex and having excessively large parameter values. A good regulariser is able to balance the variance and bias of the model, resulting in a more reliable model with improved generalisation. There are different regularisation techniques in deep learning. L2 regularisation and dropout will be explained in this sub-chapter, as these techniques are used in this research.

2.5.1 L2 regularisation

L2 regularisation is a weight regularisation technique that updates the loss function by adding a regularisation term:

$$\text{Loss function} = \text{Loss} + \lambda \sum_{i=1}^N w_i^2. \quad (3)$$

The regularisation parameter λ is multiplied by the sum of the squares of the weights w_i to form the regularisation term. L2 regularisation encourages the weights to diminish towards zero and is also referred to as weight decay as it forces the weights to take on smaller values. This prevents the weights from getting out of control and thereby avoids an exploding gradient.

2.5.2 Dropout

Another way to prevent overfitting in neural networks is to use dropout layers. In these layers, random nodes are set to the value zero; i.e. nodes and their connections are randomly dropped from the network and do not influence the prediction. The dropout rate determines the proportion of the input units to drop. Dropout limits the network's ability to adapt to the training set, thus preventing the weights from overfitting to the training data.

2.6 Convolutional neural networks

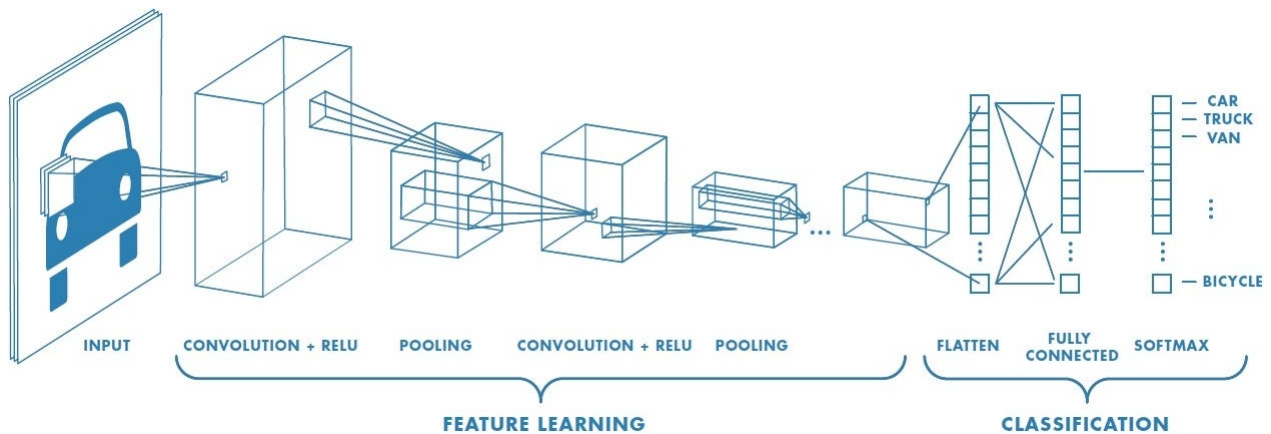


Figure 5: Example of a CNN made for the classification of vehicles with different convolutional layers and pooling layers [20].

Convolutional neural networks (CNNs) are a type of artificial neural networks that are commonly used for computer vision tasks in deep learning. CNNs have been successfully used to improve current state-of-the-art methods on many image classification, object detection, and segmentation tasks [21] and are therefore of interest in this study. They are designed to process data in the form of multiple arrays [22], which is ideal for handling images. Colour images can be represented as three 2D arrays that contain the pixel intensities of each of the

three colour channels (red, green, and blue). CNNs can successfully capture temporal and spatial dependencies of images using relevant filters. They can learn patterns in one section of an image and can recognise that pattern in other parts of the image. CNNs reduce the size of images throughout the network, making them easier to process while still retaining the essential features necessary for successful learning. The earlier layers of the CNN will be able to identify small local patterns, which will then be combined to form larger patterns in the subsequent layers. Compared to traditional artificial neural networks, CNNs can have various types of layers, such as convolutional layers, activation layers, and pooling layers, an example of a CNN with different layers is shown in Figure 5. These layers will be explained in the following sub-chapter, Chapter 2.6.1.

2.6.1 Convolutional layers

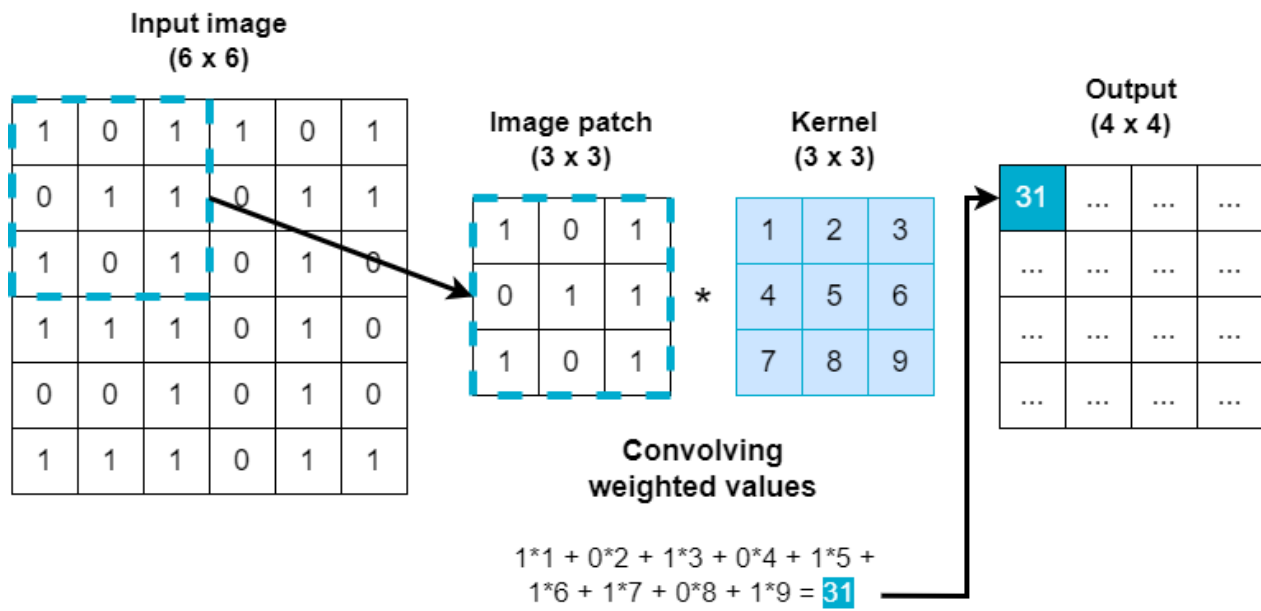


Figure 6: Example of the first step of a 3 x 3 filter convolving across an input of 6 x 6. The first value of the output is calculated by multiplying the weights element-wise in the kernel with the values in the same-sized image patch.

In a convolution layer, various filters slide, convolve, across the input data. These filters, also known as kernels, contain a collection of weights that can be learnt. The weights are the trainable parameters of the model and therefore will change during training until they reach their optimal state. Relevant features of the input images, such as horizontal lines or other specific shapes, will be represented by these kernels. Figure 6 shows an input image and a 3 x 3 kernel that convolves across the input image resulting in an output tensor, called a feature map. The value of the hidden state in the next layer, the feature map, is determined by the product of the weights in the kernel and an image patch of the same size as the kernel. The kernel moves across the entire input image until all elements of the 4 x 4 feature map are filled. This figure illustrates the reduction in the size of the feature maps; the input size

of 6 x 6 is reduced to an output of 4 x 4 after the convolutional operation. This reduction depends on the size of the filter, the stride, and the padding applied, which will be explained next. The stride is the number of pixels that the kernel shifts after each multiplication of the input image. The kernel in Figure 6 will move one step to the right to calculate the next value in the output and thus has a stride of one. When a larger stride is used, the kernel makes bigger steps during convolving, resulting in a smaller output size. Padding also influences the output size of the feature maps; however, no padding is used in Figure 6. By padding, additional pixels are added to the border of the input images. These extra pixels often have a value of zero, which is referred to as zero-padding.

Transposed convolutional layers

General convolutional layers can thus reduce the output size, but it is also possible to increase the output size by using the transpose of these layers, which is useful in upscaling images. Transposed convolutional layers apply padding to the input feature map and then perform a convolution to increase the size of the feature map. Upsampling can also be done with interpolation techniques instead of transposed convolutions. An example of this is an upsampling layer that enlarges images using the nearest-neighbour method, also used in this study.

Depthwise separable convolutional layers

Another kind of convolutional layer used in this research are depthwise separable convolutions. These layers first perform a depthwise spatial convolution followed by a pointwise convolution. A depthwise spatial convolution is a spatial convolution performed separately for each input channel. The pointwise convolution is a 1 x 1 convolution that projects the channels produced by the depthwise convolution into a new channel space [23]. This decreases the number of parameters and computations in the model in comparison to using regular convolutional layers, which reduces training time and can assist in regularising models.

2.6.2 Activation functions

Convolutional layers are commonly followed by a non-linear activation function, to add non-linearity to models, allowing it to learn and carry out complex tasks. Without this, the neural network is equal to a linear regression model. The rectified linear unit (ReLU) activation function, Figure 7, is often used in CNNs, as shown in Figure 5. This piece-wise linear function will return the input value if it is positive, and will output zero if it is not:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0. \end{cases} \quad (4)$$

ReLU's keep a lot of the characteristics that make linear models generalise well, due to their near linearity. They are easy to use, fast, and achieve good performance [24].

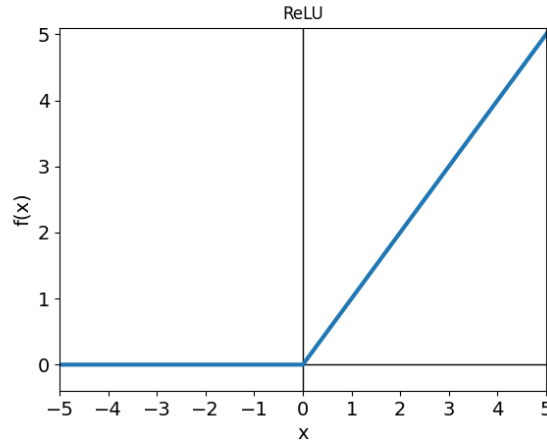


Figure 7: The ReLU activation function.

The softmax activation function is often used in the last layer of a deep learning model for multiclass problems. In this research, there are three classes, namely fibres, matrix, and voids, making it a multiclass problem. The softmax function, Equation 5, normalises the output of a neural network to a probability distribution across the predicted output classes [18].

$$\text{Softmax}(\mathbf{y})_i = \frac{e^{y_i}}{\sum_{j=1}^n e^{y_j}}. \quad (5)$$

Here, \mathbf{y} is the input vector that contains n elements, the number of classes, and y_i is the i -th element of the input vector. The elements of the output vector $\text{softmax}(\mathbf{y})_i$ add up to 1, and the probability of each class is in the range 0 to 1.

2.6.3 Pooling layers

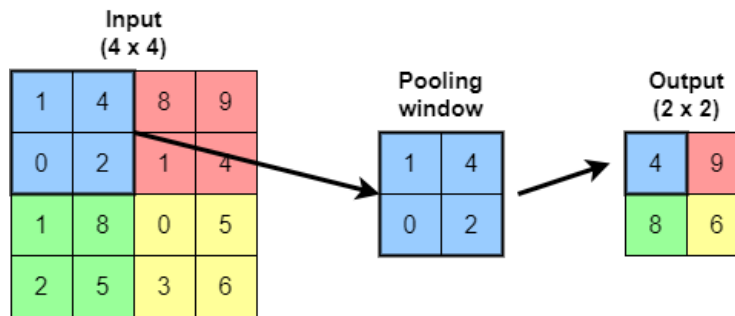


Figure 8: Example of a max pooling operation with a 2 x 2 kernel and a stride of two.

Pooling layers are used to reduce the size of a feature map, thereby decreasing the number of parameters that need to be learnt and thus the amount of computational work [15]. These layers summarise the features present in an area of the feature map created by the previous layer, making the model more robust to variations in the locations of the features in the

input image and thereby decreasing the chance of overfitting. This is done by moving a two-dimensional kernel across the feature map and summarising the features located within the area covered by the kernel. There are several pooling operations such as max pooling, average pooling, and global pooling. With max pooling, the maximum value of the region covered by the kernel is selected for the output feature map. Figure 8 shows a max-pooling operation with a 2×2 kernel and a stride of two. The first region of the input in the upper left corner that is covered by the kernel has a maximum value of four. The kernel shifts two places to the right, selecting the maximum value of nine. This procedure is repeated for the entire input.

2.6.4 Batch normalisation layers

Training deep neural networks with multiple layers can be challenging, as the inputs of each layer are altered during the training process due to changes in the parameters of the preceding layers. These ever-changing inputs lead to layers needing to constantly adapt to the new distributions. This phenomenon of changes in the distribution of inputs to layers in networks is called internal covariate shift and slows down the training process because lower learning rates are needed and the parameters need to be carefully initialised. Batch normalisation addresses this problem by normalising the layer inputs. These layers can be used in networks to increase the stability and speed of the learning process [25].

2.7 U-net

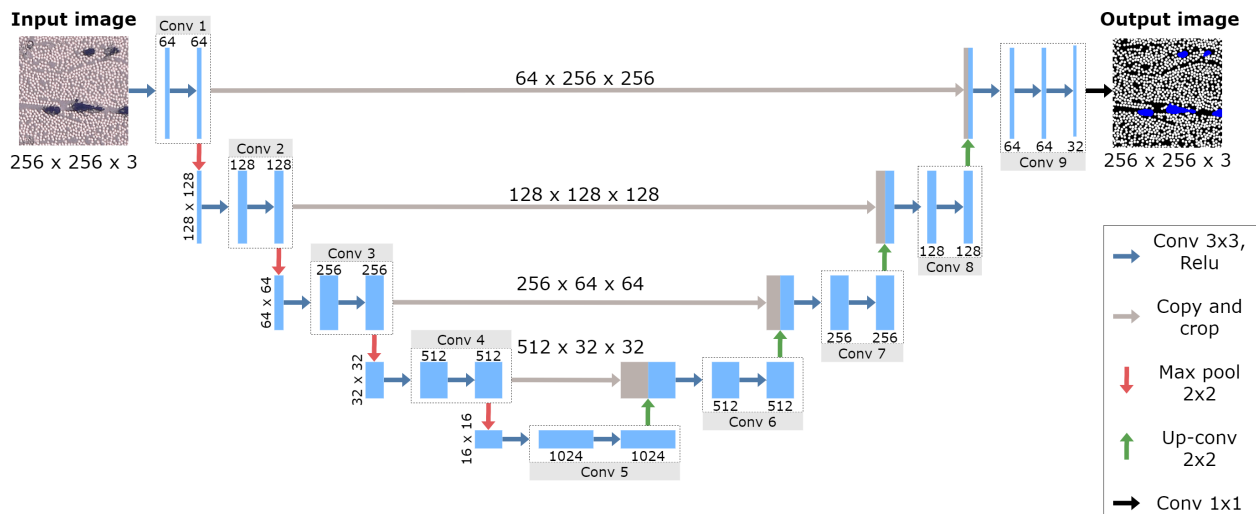


Figure 9: Original u-net model as proposed by Ronneberg et al. [26].

A u-net model is a convolutional neural network that was developed specifically for the image segmentation of biomedical images by Ronneberger et al. [26]. This architecture is created to be effective with limited training data and provides precise segmentation outcomes. The original u-net architecture follows a typical encoder-decoder architecture and is shown in Figure 9. The encoder-decoder architecture is applicable for this purpose, as it yields outputs

with the same size as the inputs. An encoder-decoder is a deep learning model that consists of two neural networks. The encoder extracts significant features from the images, while the decoder takes the extracted features and reconstructs the segmented image.

Figure 9 shows that the encoder part follows a contracting path and the decoder follows an expansive path, resulting in a u-shaped architecture. The encoder in this u-net consists of repeating convolutional layers with 3×3 kernels followed by a ReLU activation function. Then the feature maps are downsampled by using max pooling layers with a filter size of 2×2 and a stride of two. This increasingly reduces the spatial dimensions of the input and increases the number of feature maps. The decoder part of the model uses 2×2 up-convolutional layers, which now halve the number of feature maps again. The u-net architecture is characterised by doubling the number of feature maps in the decoder part of the model by concatenating the feature maps from the encoder part of the model. The concatenating method helps the network capture the positions of significant features in the images. Then 3×3 convolutions are performed, each time followed by ReLU operations. The final layer is a 1×1 convolutional layer with a softmax activation function, which gives the probability distribution over all predicted classes.

This architecture enables the network to be trained with smaller data sets than other convolutional neural networks, making it a desirable choice for this research. Furthermore, several studies demonstrated that u-net architectures are effective in addressing segmentation issues. Zhang et al. [27] showed that a u-net structure performed better than four other commonly used image segmentation methods for the segmentation of purple rapeseed leaves. Furthermore, Chen et al. [21] investigated the number of articles published from January 2016 to August 2019 on deep learning methods for cardiac image segmentation and found that most state-of-the-art segmentation methods are based mainly on CNNs that use a fully convolutional network or a u-net architecture. Therefore, the model used in this research will also be largely based on the u-net architecture.

Figure 10 visualise some feature maps within the final network, as described in Chapter 4.1, for an example microscopy image with carbon fibres and a void. The exact number of feature maps in each stage is indicated above the four feature maps shown, as it is impossible to show all feature maps in one image. As can be seen in the first stage of the u-net 32 feature maps are created by the convolutional layers in the network. This increases to 256 at the bottom of the u-net where the input image is barely recognisable. Then in the encoder part, the feature maps are decreased again at every stage and the input image is reconstructed with the relevant features captured by the model. The last convolutional layer has as many feature maps as classes present, three in this case: matrix, fibres, and voids. These three feature maps are used to create the final segmented output where each pixel indicates to which of the three classes it belongs.

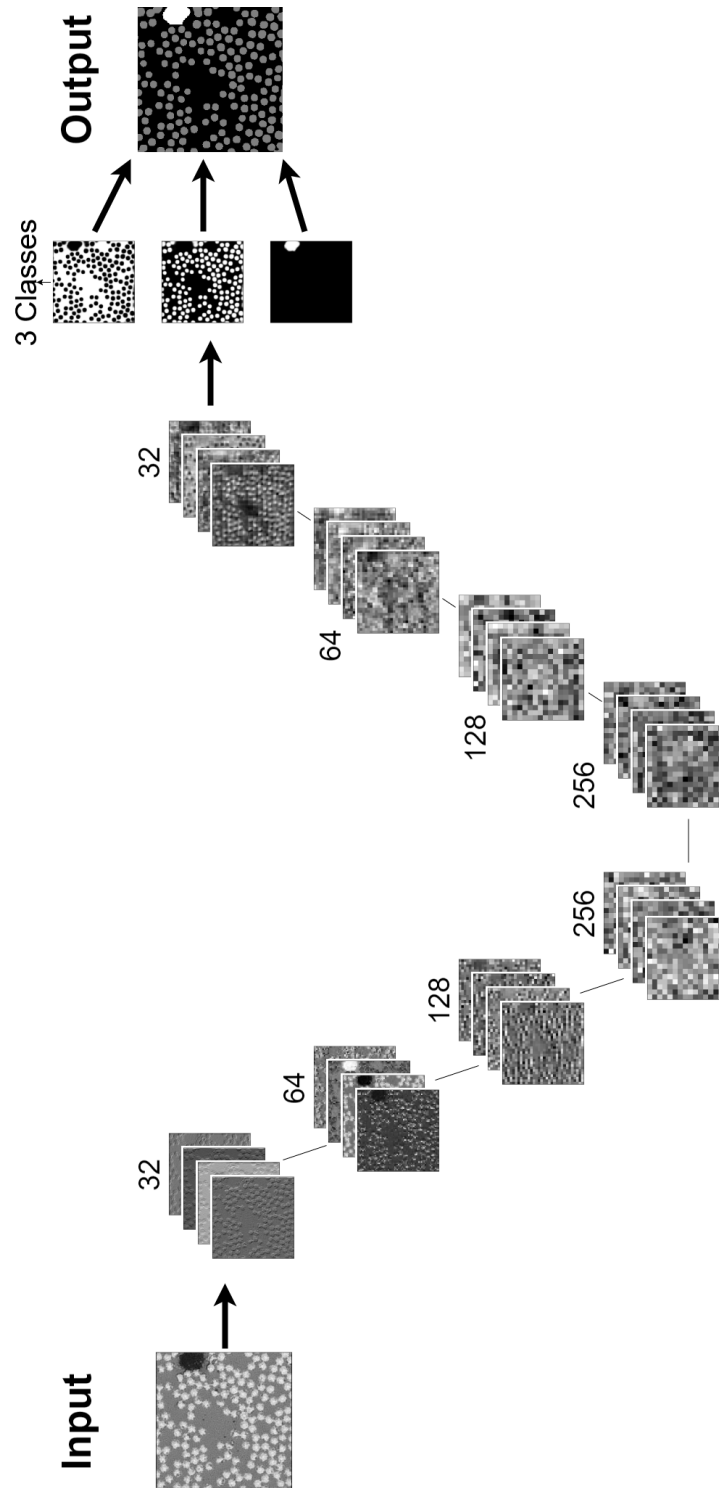


Figure 10: Overview of the feature maps in a trained u-net model for the segmentation problem of the fibres and voids. Only four examples of possible feature maps are shown in this image for clarity, the right number of feature maps is indicated above each stage in the u-net.

3 Data

In this chapter, the methods used to create different data sets with training data are explained. A schematic overview can be seen in Figure 11. First, 30 micrographs are obtained and cut into images of 256 x 256 pixels. These square images are divided into 14 image sets with similar-looking images that can undergo the processing steps in the mask generation process with the same parameters. During mask generation, first fibres are identified in the images, and then voids are detected, as described in Chapter 3.3. Three different training data sets are made with these images and their corresponding masks. Additionally, a test set is created with images not included in the training data. The training data sets will be augmented with seven different methods as explained in Chapter 3.5. Then different u-net models will be trained with the augmented data sets and the non-augmented data sets to see the difference using the metrics during evaluation. Lastly, the trained models are tested with the test data set. These results are also evaluated with evaluation metrics; the accuracy and the Intersection over the Union (IoU).

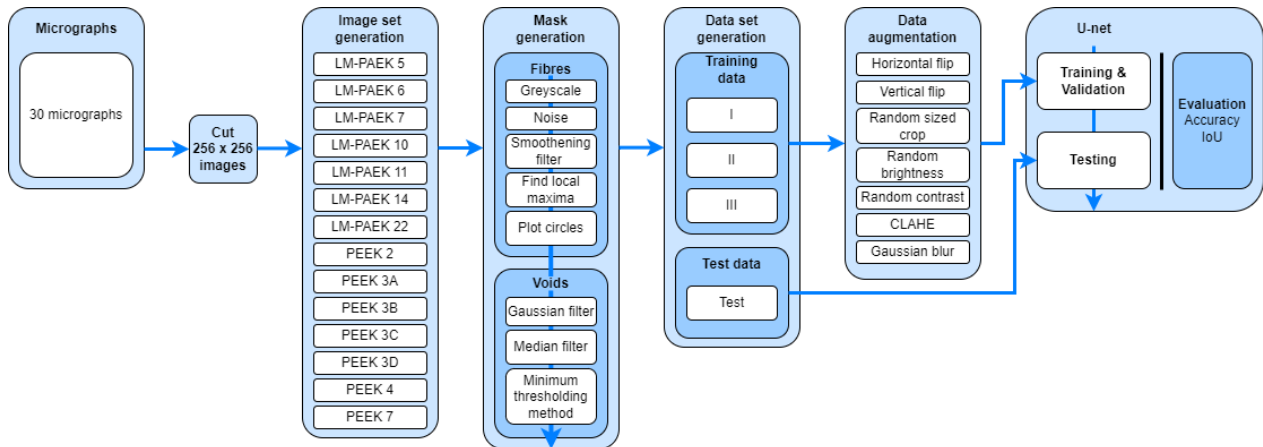


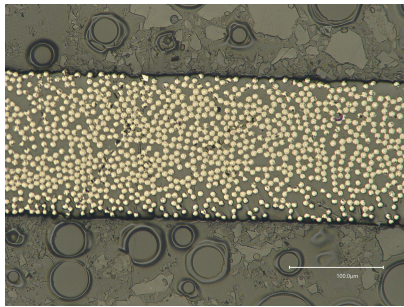
Figure 11: Schematic overview of the data generation steps needed to train and test the u-net model.

3.1 Micrographs

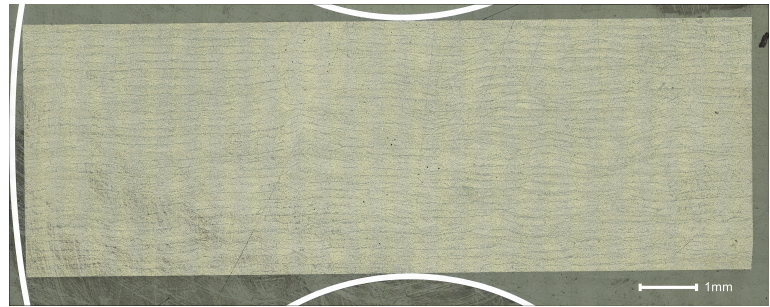
First, data needs to be obtained to train the u-net model. Different micrographs from cross-sections of FRPs are needed that include voids. It is important that the data is as diverse as possible, so that the model can learn from different situations during training to be able to make better predictions.

For this research, 30 different carbon fibre reinforced polymers (CFRP) micrographs were collected from the ThermoPlastic composites Research Center (TPRC). Micrographs were made by cutting small samples from composite laminate specimens with a diamond saw. These samples were embedded in Struers EpoFix epoxy. When hardened, the surfaces of the cross-sections were polished with a Struers Tegramin 30 machine to obtain clear micrographs with the Keyence vhx-7000 confocal microscope. The CFRPs used in this research are

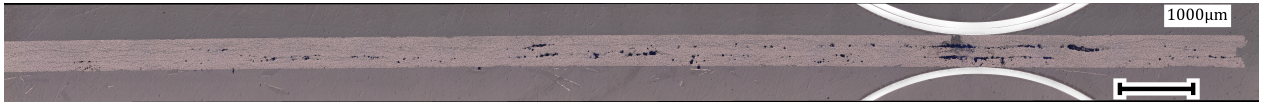
made of carbon-reinforced low-melting polyaryletherketone (LM-PAEK) or carbon-reinforced polyetheretherketone (PEEK). However, the type of matrix material and the laminate details of the micrographs are irrelevant for this study, but are explained in more detail in Appendix A. The presence of voids in the micrographs is relevant. Unfortunately, not all micrographs contain voids. Figure 12 shows examples of micrographs of different laminates that are used. Figure 12c clearly shows the voids indicated by the darker spots; however, the other examples barely contain any voids.



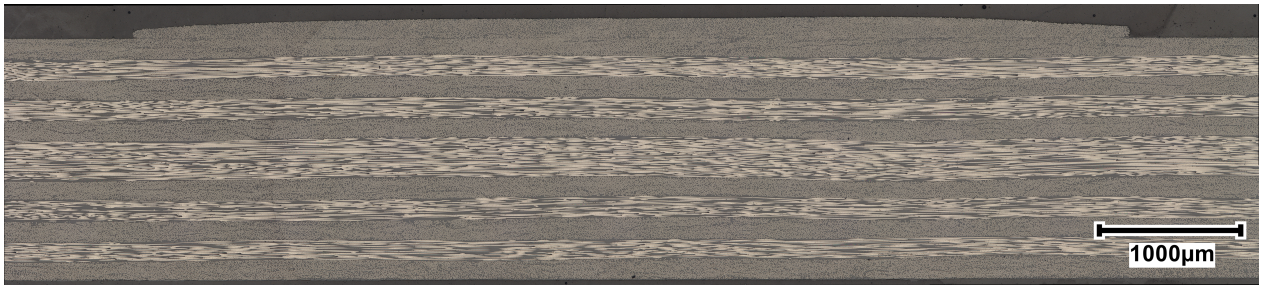
(a) Micrograph (“specimen_37_left”, 2048 x 1536 pixels) of a one layer tape of LM-PAEK specimen without voids [28].



(b) Stitched micrograph (“G02_3”, 20495 x 8121 pixels) of a PEEK specimen with a $[0]_{16s}$ lay-up.



(c) Stitched micrograph (“2_6_2_R”, 15701 x 1222 pixels) of a friction tested PEEK specimen with visible voids [29].



(d) Stitched micrograph (“0_press_375_R2_300x”, 16235 x 3638 pixels) of a LM-PAEK specimen without voids and a $[0/90]_{6s}$ lay-up with an extra top-layer [30].

Figure 12: Examples of different micrographs used in this research.

3.2 Image sets

To create training data from the 30 micrographs, large images need to be cut into smaller images. The micrographs were cropped to 256 x 256 pixels with a Python script using the `itertools` module and the Python Imaging Library (PIL). This size was chosen because larger sizes of images would drastically increase the number of computations and memory needed to train the models. From a set of 30 micrographs, 1582 training images were obtained. The

locations of where the images are cut from the micrograph are stored in the name of each separate image, since each image is labelled in the form NAME_Y_X. Where NAME is the original name of the micrograph and X is the starting location on the X-axis of the original micrograph from where that specific image is taken. Y is the starting location looking at the Y-axis of the original image from which the training image was cut. So, the example image bounded by the blue dotted line in Figure 13 is called EXAMPLE_0_0, since the cut started at the top of the image at the location (0,0). These names can be found in the data sets which are made available online [31].

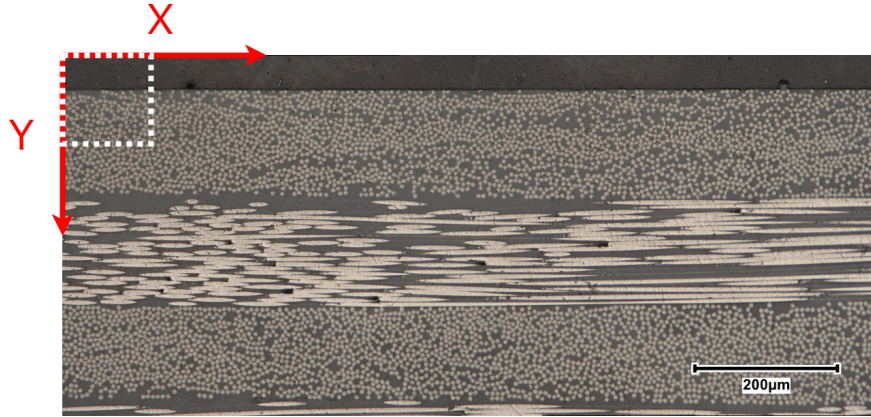
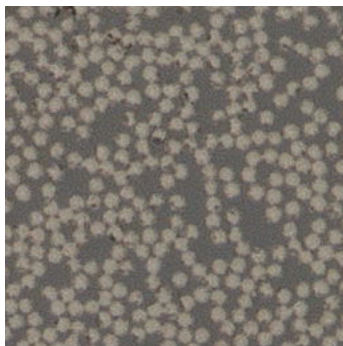
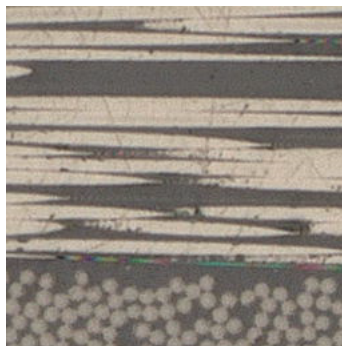


Figure 13: Example of how an image is cut from a micrograph. The blue dotted square indicates a 256 x 256 image called EXAMPLE_0_0.

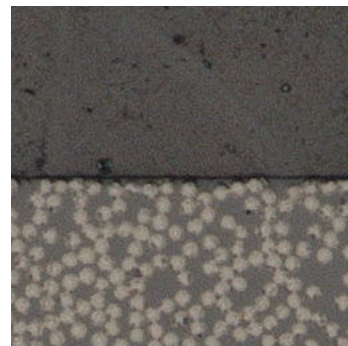
The training images are cut from the relevant regions of the micrographs, with the fibres in the zero direction, which point outwards of the paper as shown in Figure 14a. Therefore, images with fibres in directions other than the zero direction or images showing a clear edge of the composite are not used for training data, as shown in Figures 14b and 14c. This is because these images are not useful for determining the void and fibre volume fractions, and thus only make the learning process of the u-net models unnecessarily difficult.



(a) Useful training image of 256 x 256 pixels.



(b) Image showing both a 0° and 90° layer, not used as a training image.



(c) Image showing the edge of the laminate, not used as a training image.

Figure 14: Three different images of 256 x 256 pixels that are cut from the micrograph shown in Figure 12d.

Table 1: Overview of the 14 created image sets with their image source, fibre radius, total number of images and images containing voids.

Image set	Image source	Fibre radius [pixels]	Number of images	Number of images with voids
LM-PAEK 5	2	5	100	0
LM-PAEK 6	5	6	100	9
LM-PAEK 7	1	7	16	2
LM-PAEK 10	2	10	250	0
LM-PAEK 11	5	11	100	7
LM-PAEK 14	3	14	100	0
LM-PAEK 22	5	22	100	4
PEEK 2	4	2	100	42
PEEK 3A	1	3	356	305
PEEK 3B	1	3	68	42
PEEK 3C	1	3	76	58
PEEK 3D	4	3	100	19
PEEK 4	4	4	100	36
PEEK 7	1	7	16	2

All useful images cut from the 30 micrographs are divided into different image sets that contain images that appear the same, that is, a similar fibre size and similar lighting conditions. Therefore, these sets of images can undergo the same post-processing steps, as explained in Chapter 3.3. Thus, the images can be handled in batches and do not need to be analysed one by one, which facilitates mask generation by speeding up the process. Each image set is labelled with the material and the size of the fibres in pixels of the images. For example, the image set PEEK 4 includes images of fibres with a radius of four pixels that consist of PEEK matrix material. When multiple sets of images are taken from the same specimen with the same fibre size, a letter is added to the end of the name in alphabetical order, for example, PEEK 3A. An overview of all the image sets created is shown in Table 1. This overview provides the total number of images per image set and the proportion of them that contain voids, as well as the fibre radius and their original image source, as explained in Chapter 3.1.

3.3 Mask generation

The images from the different image sets needed to be annotated before they could be used for training purposes. In this annotation process, each pixel of all images is labelled to indicate to which class it belongs. Label encoding is used to represent the matrix material with a value of zero, fibres with a value of one, and voids with a value of two. These annotated images are saved separately and are used as ground truths (segmented output) for the original input images. The annotation process is performed using ImageJ [32], which is an open-source image processing programme, and the Python scripts developed in this study. Note that these annotated images initially appear black, due to the use of the discrete labels

zero, one, and two for each pixel. To make the masks visible, the labels are converted to clearly distinguishable pixel values with the help of the numpy, PIL, and TensorFlow packages in Python. These converted images are presented in this report.

3.3.1 Fibres

The centres of all the fibres present in each image are found using image processing functions using ImageJ. The images in a given set are evaluated together using the same parameters, which is possible because the images in a set share similar features. The centres of the fibres are extracted by using the “Find maxima” function in ImageJ. This function determines the local maxima in the pixel values present in an image, and the coordinates of these maxima can be saved in a list for each image for further use. To achieve acceptable results using the “Find maxima” function, the sets of images needed to be preprocessed to remove noise.

Table 2: ImageJ parameters that are used in the mask-making process of different subsets of data.

Image set	Smooth filter [times]	Prominence	greyscale	Noise
LM-PAEK 5	6x	>3	Yes	No
LM-PAEK 6	3x	>2	Yes	No
LM-PAEK 7	5x	>10	Yes	No
LM-PAEK 10	10x	>5	Yes	No
LM-PAEK 11	10x	>5	Yes	No
LM-PAEK 14	15x	>20	Yes	Yes
LM-PAEK 22	10x	>15	Yes	Yes
PEEK 2	1x	>2	Yes	No
PEEK 3A	3x	>2	Yes	No
PEEK 3B	4x	>2	Yes	No
PEEK 3C	5x	>2	Yes	No
PEEK 3D	4x	>1	Yes	No
PEEK 4	5x	>2	Yes	No
PEEK 7	7x	>15	No	No

Blurring images before using the “Find maxima” has been shown to help increase the accuracy of local maximum detection [33]. This is done using the “smooth” filtering function in ImageJ, which blurs the image by replacing each pixel with the average value of the 3 x 3 neighbourhood of that specific pixel [34]. A 3 x 3 filter is a commonly used size for this filtering method and is a fixed parameter in the ImageJ program. This operation can be executed multiple times on one image, depending on the specific needs of the particular set of images. The specific number of times this filtering function has been used for each of the different sets of images is shown in Table 2. Figure 15 illustrates how blurring the image improves the number of correctly predicted fibre centres. The predicted centres of the fibres are indicated by a white cross with a yellow circle in the middle. The prominence, a param-

eter of the “Find maxima” function, has been fixed to show the effect of multiple executions of the smoothing filter.

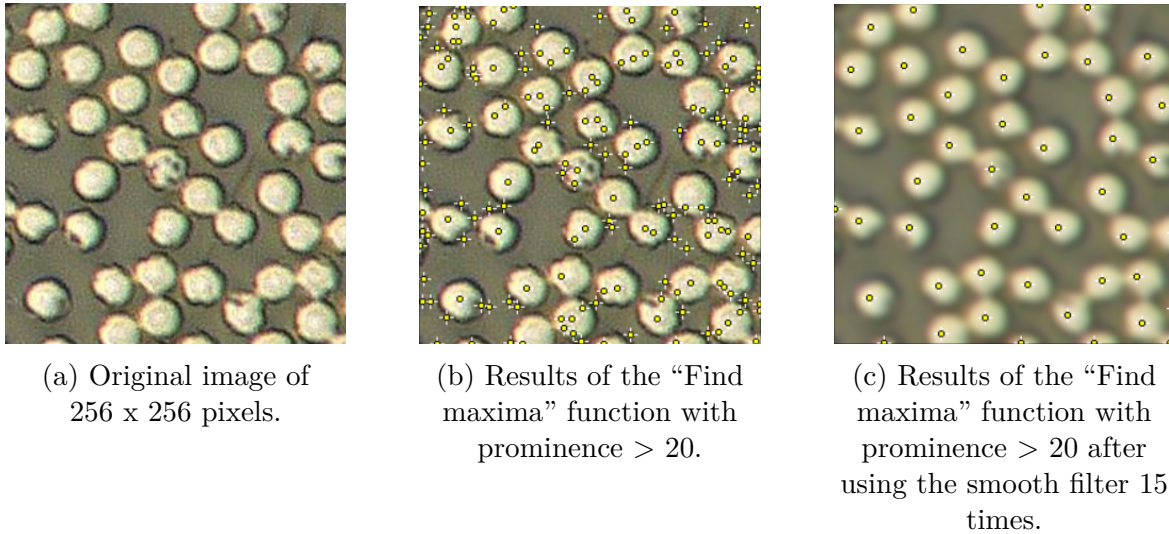


Figure 15: Influence of blurring the image before searching for local maxima.

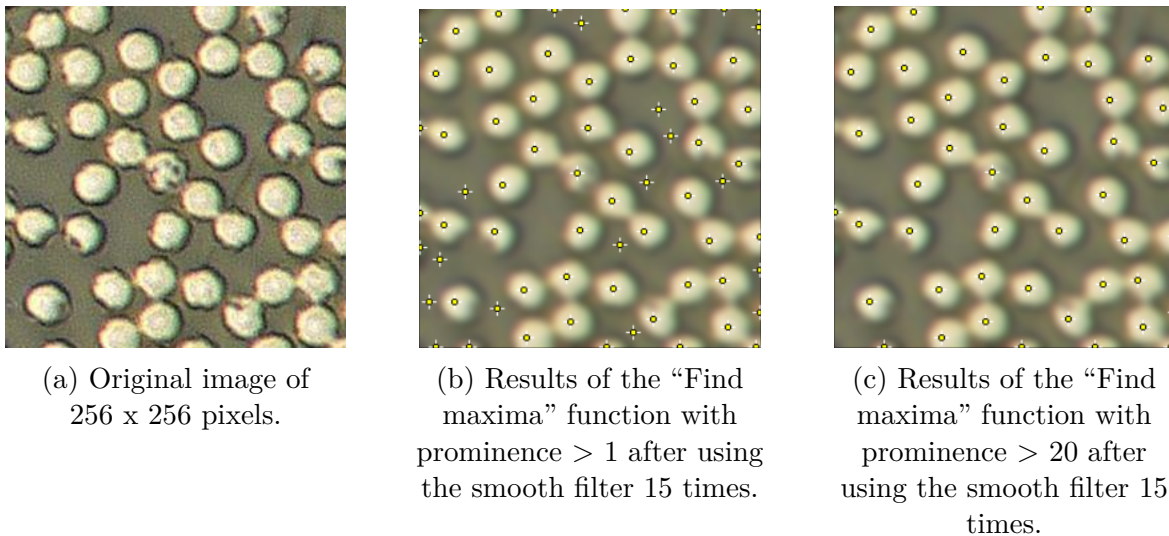


Figure 16: Influence of the prominence parameter while searching for local maxima.

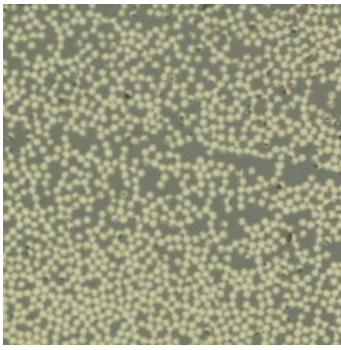
Furthermore, the “Prominence” parameter is a noise tolerance parameter that defines by what values maxima should stand out from its surroundings to become a local maximum. Thus, a local maximum does not contain any point with a value higher than its maximum in the area specified by this parameter [34]. This means that when this value increases, fewer local maxima are detected, and this value is generally higher for images that contain fibres with a larger radius. The influence this parameter has on the process is shown in Figure 16, where the number of times the smooth filter is used is fixed. Thus, this parameter of the

“Find maxima” function must be carefully chosen. The final values selected for each set of images are shown in Table 2.

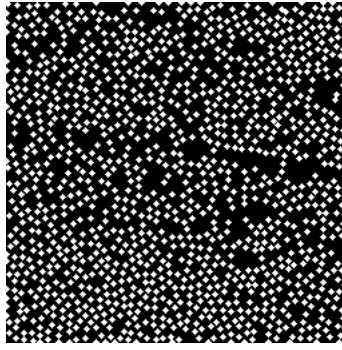
Lastly, to decrease the number of fibres that are incorrectly predicted, noise can be added. Noise adds unevenness to the images. This first increases the local maxima, but after using the smoothening filter, fewer local maxima are found at locations where no fibres are present. This was done using the “Add noise” function in ImageJ, This function adds random noise to the image, which is Gaussian distributed with a mean of zero and a deviation of 25 [34]. Two image sets with large fibre sizes showed better results when this noise was first added to the images, as can be seen in Table 2. To furthermore increase the number of correctly predicted fibre centres, the subsets of images could be converted to greyscale. The sets of images in which this leads to better performance are shown in Table 2.

All these different parameters are tuned for each image set based on trial and error. In this tuning process, a representative image of each subset was used to check the accuracy of detecting all fibres while minimising false detection of fibres. Hereafter, these parameters have been applied to each image of that subset, while carefully monitoring the accuracy of correctly detecting fibre centres. All the parameters used in each set of images can be found in Table 2.

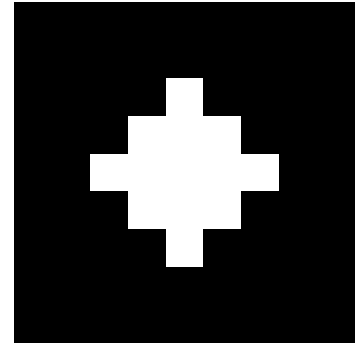
Now that the locations of the centres of the fibres are known, these centres will be used to create the masks. It is assumed that the fibres are perfect circles that have the same radius in each set of images. The radius of these fibres is analysed by counting the radius in pixels in the image for some fibres and averaging this number. However, for the mask-making process, the radius will need to be a natural number, as a pixel either belongs to a fibre or not, in the mask. Due to the chosen image size of 256 x 256, this limits the accuracy of the masks. This is especially the case for data that contains fibres with smaller radii, as there will be fewer pixels per fibre, which makes it more difficult to closely resemble a circle. This is illustrated in Figure Figure 17. Three example images are shown here from three different image sets with fibre radii of two, three, and 22 pixels. It can be seen that smaller radii result in a diamond-shaped fibre, Figures 17c and 17f, while a larger radius comes closer to a circle-shaped fibre, Figure 17i. It should be noted that the fibre size of Figure 17a and 17d does not differ much, but since this process is restricted to natural numbers for the fibre radii, the total area covered by pixels belonging to the fibres changes significantly. The radii used for the masks of each image set are already given in Table 1.



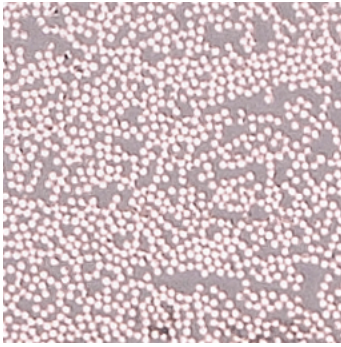
(a) Original image.



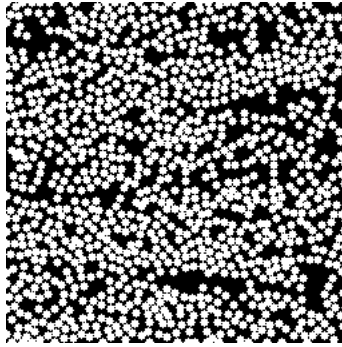
(b) Mask with a fibre diameter of five pixels.



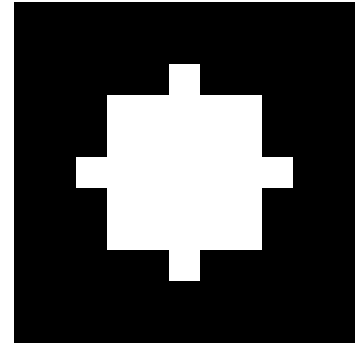
(c) "Circle" with a diameter of five pixels.



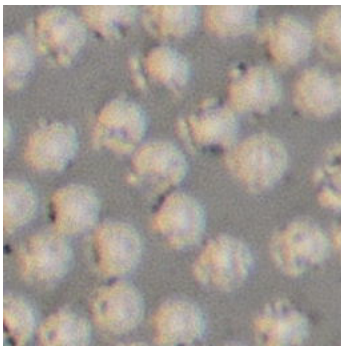
(d) Original image.



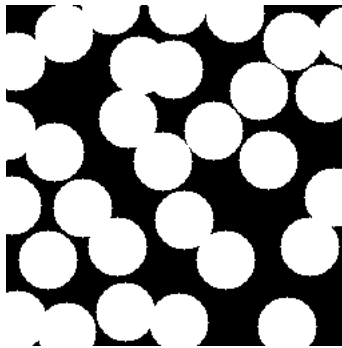
(e) Mask with a fibre diameter of seven pixels.



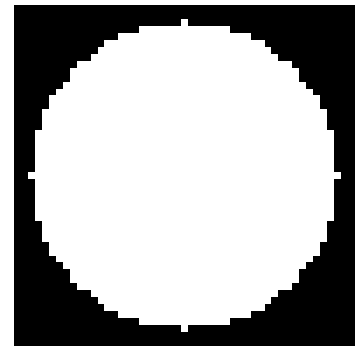
(f) "Circle" with a diameter of seven pixels.



(g) Original image.



(h) Mask with a fibre diameter of 45 pixels.



(i) "Circle" with a diameter of 45 pixels.

Figure 17: Example images with their corresponding mask and fibre size.

3.3.2 Voids

The masks now contain fibres and matrix material, but there are still no voids. Therefore, all images that contain voids will be sorted and will undergo a separate thresholding process. First, the images will be converted to greyscale images. Then, a Gaussian filter, also called a Gaussian kernel, will be applied to smooth and reduce noise in the images. This filter modifies the input by a convolution with a Gaussian function. This process is shown in Figure 18, where one step of this process is visualised. The two-dimensional Gaussian function is given in Equation 6.

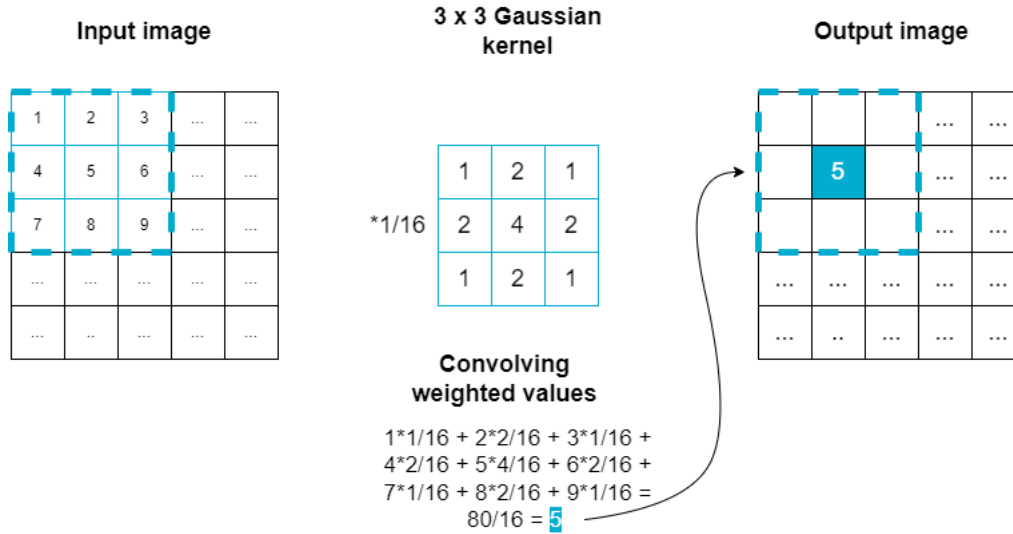


Figure 18: A discrete Gaussian 3 x 3 kernel shown in the middle that is used to transform the input image (left) into an output image (right) by convolving the weighted values from the kernel over the input image to form the centre's pixel value as shown in the output image.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (6)$$

where x and y are the x- and y-coordinates respectively and σ is the standard deviation of the Gaussian distribution. This function can then be used to calculate a Gaussian kernel. For getting a 3 x 3 kernel, the x- and y-coordinates will range from minus one to one, with (0,0) being the centre of the kernel. Thus, the kernel has rotational symmetry, as shown in Figure 18. This distribution is assumed to have a mean of zero and shows that the kernel coefficients closer to the centre of the kernel have higher values. To maintain the Gaussian nature of the filter, the filter size should increase with an increasing standard deviation. Therefore, the kernel size, S_{kernel} , along each axis will be calculated based on the chosen standard deviation σ ,

$$S_{\text{kernel}} = 2r + 1, \quad (7)$$

with r the fibre radius:

$$r = t\sigma, \quad (8)$$

where the parameter t thus indicates when to truncate the filter; at how many standard deviations. This parameter is set to four in this research, and thus only the standard deviation needs to be chosen per image set. The results of applying a Gaussian filter with different σ values of one and three to an image from the data set are shown in Figure 19. The values chosen for σ for each set of images are shown in Table 3. Note that the image sets containing no voids are not present in this table.

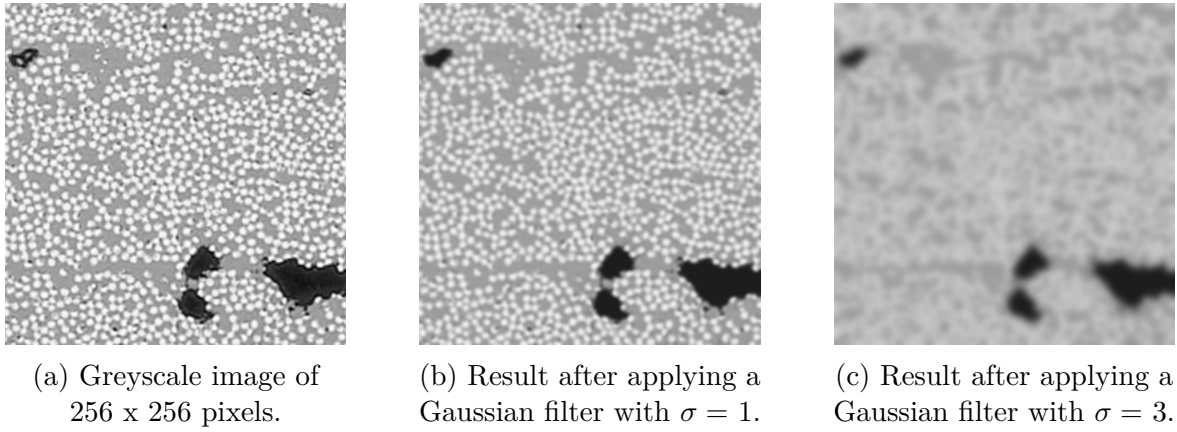


Figure 19: Influence of applying a Gaussian filter to a greyscale image.

Table 3: The different parameters used for detecting voids in the annotation process for the different image sets.

Data subset	Gaussian σ	Median filter size
LM-PAEK 6	2	3
LM-PAEK 7	3	3
LM-PAEK 11	3	3
LM-PAEK 22	3	3
PEEK 2	1	1
PEEK 3A	3	3
PEEK 3B	3	3
PEEK 3C	3	3
PEEK 3D	3	3
PEEK 4	3	3
PEEK 7	3	3

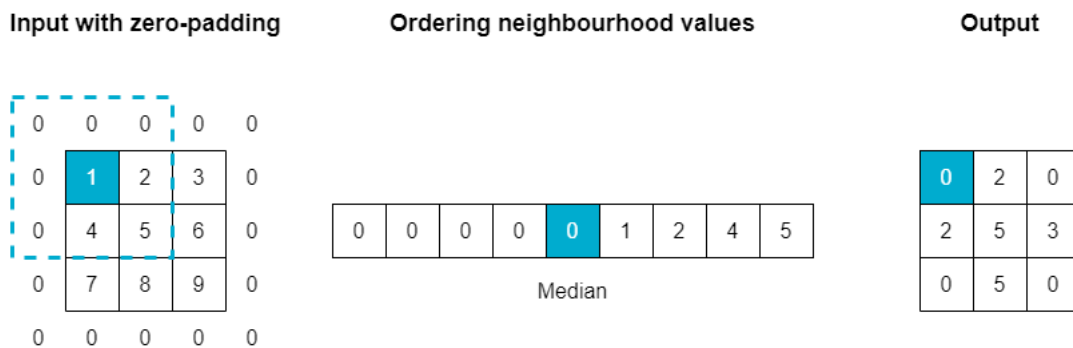


Figure 20: Process of applying a 3 x 3 median filter to a 3 x 3 image.

Next, a median filter will be applied to the already filtered image to remove noise. This filter replaces a pixel value with the median value of its neighbouring pixels. Again, the filter size

has to be chosen for each specific set of images by trial and error. An example with a 3×3 filter is shown in Figure 20. Here, a 3×3 image containing pixels with values ranging from 1 to 9 is used. To be able to slide the filter, pixel by pixel, over the entire image, zero-padding will be applied. This is a way to extend the borders outside the image with zeros. Then starting at the top left corner, all values inside the filter are sorted from smallest to greatest. The median of these values is then taken as a new value for the output. This is then done for all pixels in the image. An example of applying a median filter of two different sizes (3×3 and 10×10) is shown in Figure 21. As can be seen, noise is removed around the voids. The filter size chosen for each set of images is also shown in Table 3. Figure 23 shows the process of getting from an image to a ground truth mask including voids. After the preprocessing steps explained here, the resulting image will look like Figure 23b for the original image shown in Figure 23a.

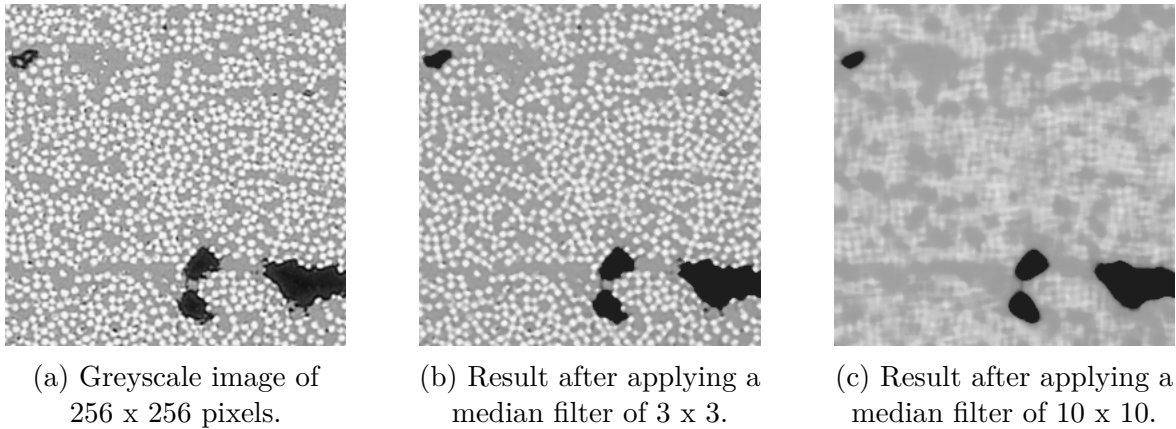


Figure 21: Influence of applying a median filter to a greyscale image.

After preprocessing the images for detecting voids, a threshold algorithm will be used. The chosen thresholding method is based on the minimum method. For this, the histogram of the input images is computed and smoothed until there are only two maxima. The minimum value between these two maxima is then the threshold value [35]. This threshold value determines which pixels will belong to voids and which pixels will not belong to voids. Figure 22 shows a training image with voids and its appearance after applying the two filters. This image, Figure 22b, is used to calculate the histogram of the intensities of the pixels, which is shown in Figure 22c. As can be seen, there are already two main peaks; one below 50 indicating the darker pixels belonging to voids and one between 150 and 200 indicating the grey pixels of the fibres and matrix material. The minimum between those two maxima, after smoothing the histogram, is calculated to be 106, which will be the threshold value. All pixels with a value less than or equal to this threshold value will be labelled as voids (label 2) in the masks already made with fibres and matrix material, see Figure 23d, to create the final ground truth mask with voids. The entire process is shown in Figure 23.

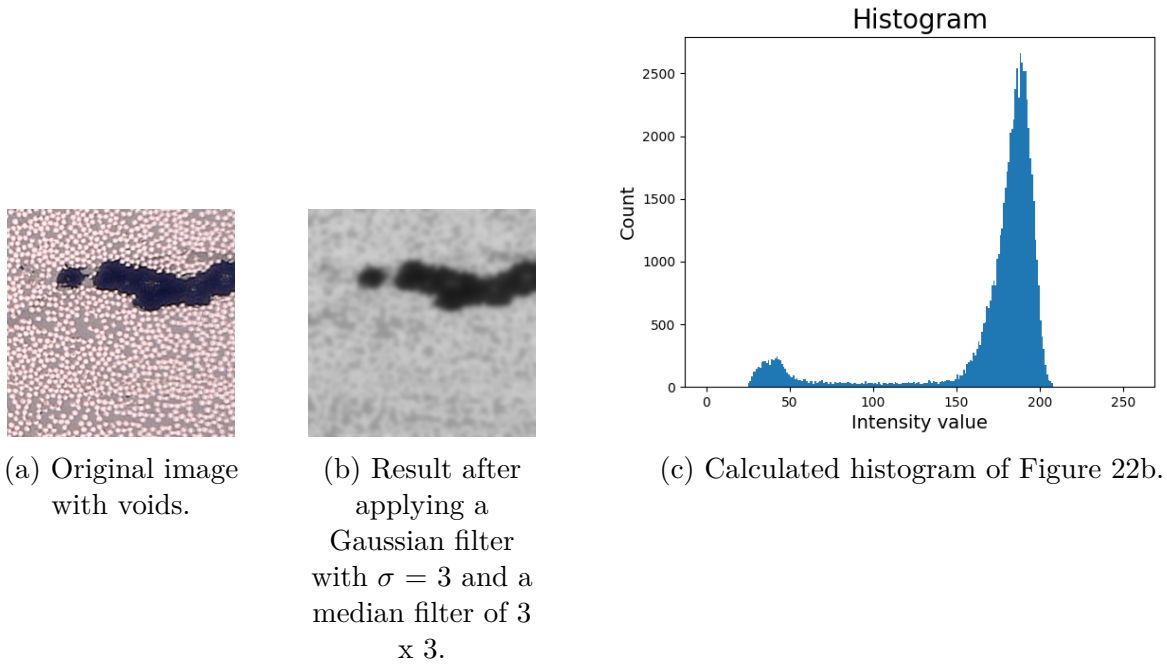


Figure 22: Process of using thresholding based on the minimum method. The calculated histogram will be smoothed until there are two maxima. The minimum value between this two maxima is 106, which will be the threshold value.

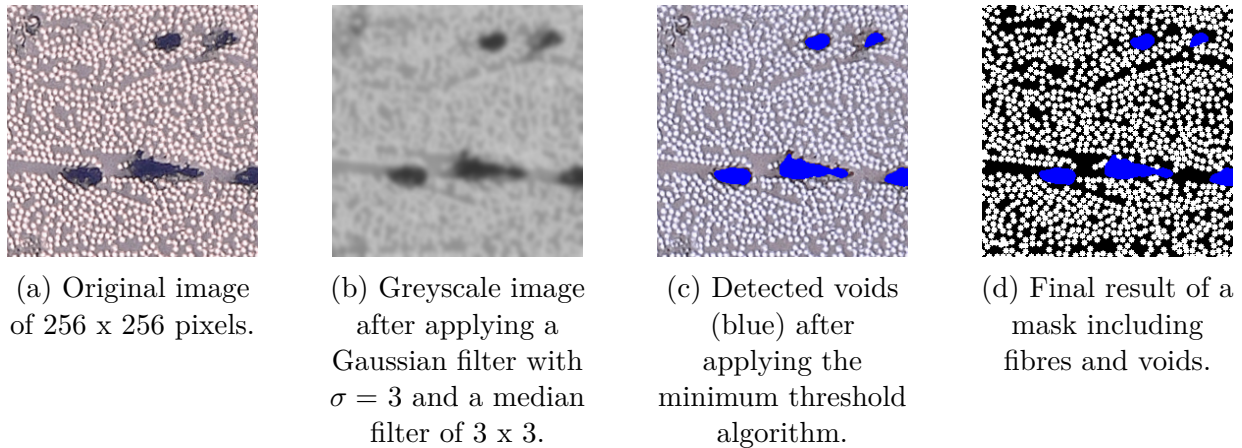


Figure 23: The process of creating the ground truth masks including voids, fibres and matrix material.

3.4 Data sets

Different data sets have been created using the images and the corresponding masks of the image subsets of Chapter 3.2. Three data sets are used to train the u-net model to achieve good performance results. An overview of the created data sets is shown in Table 4. The data set I is created from the first available image sets and is used to find out if it is possible for a u-net model to recognise fibres, voids, and matrix material. Data set II and data set III are created to include more diverse data. Finally, a smaller data set is assembled to test the models with new data that is not included in the training data. This is done to mimic how the model performs when receiving images of a new micrograph of composite material. The structure of these four data sets is explained further in this chapter, and more detailed explanations can be found in Appendix B. All data sets can be found online at the 4TU.ResearchData data repository [31].

Table 4: Overview of the three training data sets and the test data set.

Data set name	Number of images	Purpose
I	500	First tests, with first images available.
II	500	Adds a different fibre size to the training data.
III	1000	Adds even more diversity to the training data.
Test	32	Data set to test a trained model with completely new images.

3.4.1 Data set I

This data set is created with the first available images and consists of 500 training images, from three different image sets, as shown in Table 5. All training images come from PEEK specimens with a fibre radius of three. These image sets are chosen because they contain a lot of voids. As can be seen in Table 5, 81% of the training images include voids. Examples of these training images are shown in Figure 24.

Table 5: Composition of data set I from three different image set.

Image set	Number of training images	Number of images with voids
PEEK 3A	356	305
PEEK 3B	68	42
PEEK 3C	76	58

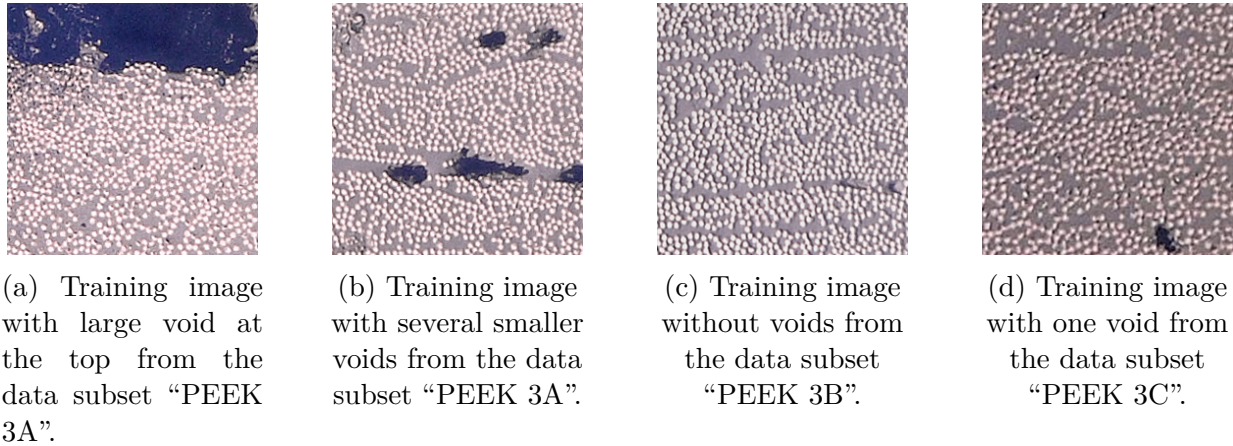


Figure 24: Examples of training images from data set I.

3.4.2 Data set II

To include images with a different fibre size, data set II is created. This data set contains again 500 training images with their ground truth masks. For this data set, 250 images and their corresponding masks from data set I are reused, as can be seen in Table 6. Additionally, 250 new 256 x 256 images were selected from the LM-PAEK 10 image set. Note that these 250 images unfortunately do not contain voids, so only 41% of the training images now contain voids. Some examples of images in this data set are shown in Figure 25.

Table 6: Composition of data set II from three different image sets.

Image set	Number of training images	Number of images with voids
PEEK 3A	174	146
PEEK 3C	76	58
LM-PAEK 10	250	0

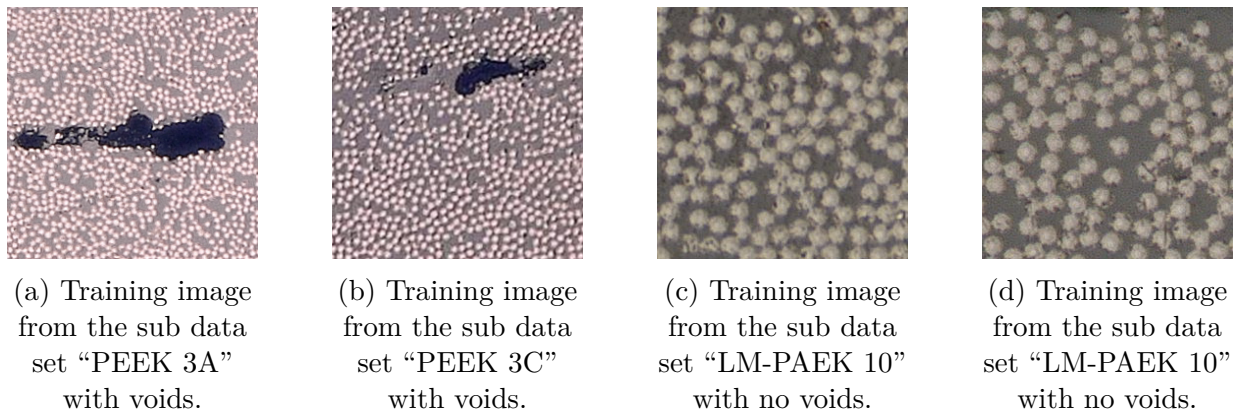
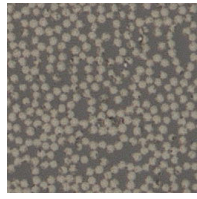
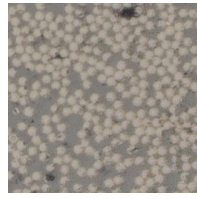


Figure 25: Examples of training images from data set II from different image sets.

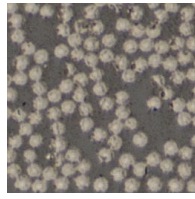
3.4.3 Data set III



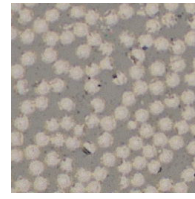
(a) Training image from the sub data set “LM-PAEK 5.”



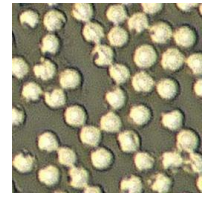
(b) Training image from the sub data set “LM-PAEK 6” with a void.



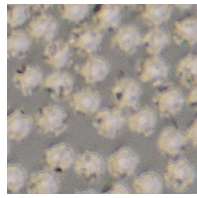
(c) Training image from the sub data set “LM-PAEK 10” with no voids.



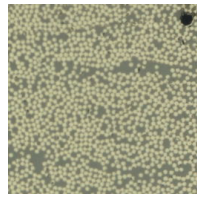
(d) Training image from the sub data set “LM-PAEK 11”.



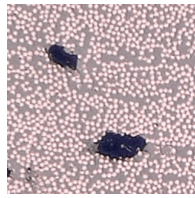
(e) Training image from the sub data set “LM-PAEK 14”.



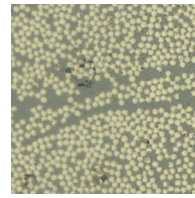
(f) Training image from the sub data set “LM-PAEK 22”.



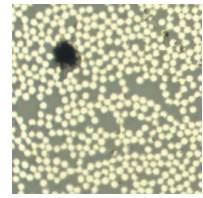
(g) Training image from the sub data set “PEEK 2” with a void.



(h) Training image from the sub data set “PEEK 3A” with voids.



(i) Training image from the sub data set “PEEK 3D” with voids.



(j) Training image from the sub data set “PEEK 4” a void.

Figure 26: Examples of training images from data set III from different image sets.

To create a more diverse and larger data set, more images from different image sets were selected to create a data set of 1000 images. All these image sets used in this data set are given in Table 7, as well as the number of images per set. An attempt was made to create ten distinct groups of 100 training images each. However, 11 image sets were used in the end, but the PEEK 3A and 3B image sets appear similar. They just have slightly different parameters in the post-processing process as explained in Chapter 3.3 and can therefore be seen as a distinct group of images. In total, this data set contains 1000 images, of which 19.1% contain voids. However, because of the unpredictability of the occurrence of voids in the laminates, these voids are not evenly spread over the data subset. Example images of each of the ten main different subsets consisting of 100 images are shown in Figure 26 and a detailed overview of this data set can be found in Appendix B.

Table 7: Composition of data set III from 11 different image sets.

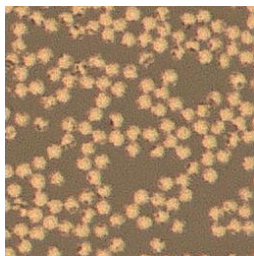
Image set	Number of training images	Number of images with voids
LM-PAEK 5	100	0
LM-PAEK 6	100	9
LM-PAEK 10	100	0
LM-PAEK 11	100	7
LM-PAEK 14	100	0
LM-PAEK 22	100	4
PEEK 2	100	42
PEEK 3A	75	60
PEEK 3B	25	14
PEEK 3D	100	19
PEEK 4	100	36

3.4.4 Test data set

This small test data set is not used for training the u-nets, but is used as a test case to illustrate a real-life example of what would happen when the model needs to analyse a new micrograph. This data set contains 32 images originating from two different image sets. As can be seen in Table 8 18.75% of the images contain voids. This is comparable to the number of images with voids in the data set 1000. Examples of these images with and without voids are shown in Figure 27.

Table 8: Composition of the test data set from two different image sets.

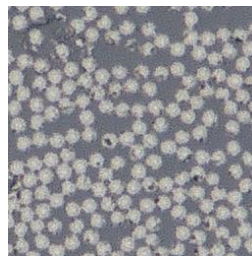
Image set	Number of training images	Number of images with voids
LM-PAEK 7	16	2
PEEK 7	16	4



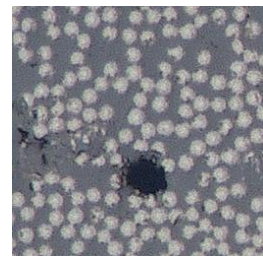
(a) Image from the sub data set “LM-PAEK 7”.



(b) Image from the sub data set “LM-PAEK 7” with void.



(c) Image from the sub data set “PEEK 7”.



(d) Image from the sub data set “PEEK 7” with void.

Figure 27: Examples of images from the test data set from the two different image sets.

3.5 Data augmentation

The created data sets lack diversity because they mainly consist of smaller images cropped from the same original micrograph. When only a few training images are available, data augmentation can be essential to teach a network the desired invariance and network properties and prevent overfitting [36]. Therefore, it will be investigated whether data augmentation can be used to improve the performance of trained models. Data augmentation is a way to intentionally increase the amount and diversity of data from the original data, by slightly transforming the data and preserving the corresponding output labels [37, 38].

Simple data augmentation techniques, summarised in Table 9, will be used to transform training images. The seven transformations are applied with a probability of 50%, which means that not all transformations are applied to every image. The transformed images and their corresponding transformed masks with a size of 256 x 256 will be used as input data for the u-net model. As an example of what these seven transformations can do, they are applied to one example image and are applied separately with a probability of 1, as can be seen in Figure 28. The transformations will be explained in more detail in the remainder of this chapter.

Table 9: The seven transformations and their parameters. All transformations have a probability of 0.5 to be applied to one image.

Methods	Parameters
Horizontal flip	-
Vertical flip	-
Random sized crop	Crop size between 128 and 250 pixels
Random brightness	Max $\pm 20\%$
Random contrast	Max $\pm 20\%$
CLAHE	Clip limit between 1 and 4, tile size 8 x 8
Guassian blur	Kernel size between 3 and 7

Geometric transformations

It is of importance that the augmented data is generated in a way that represents a more comprehensive set of possible data. Therefore, important features of the fibres should remain visible, and it is chosen to perform some geometric transformations that do not influence the circular shape of the fibres. So, no elastic transformations are applied, but the images can be flipped horizontally and vertically. A random sized crop can also be applied that crops a random square part of the input image of a size between 128 and 250 pixels. This does not influence the output size of the images, as it is resized to 256 x 256 pixels. This results in different sizes of fibres, since the size of the fibre will increase when a crop is made.

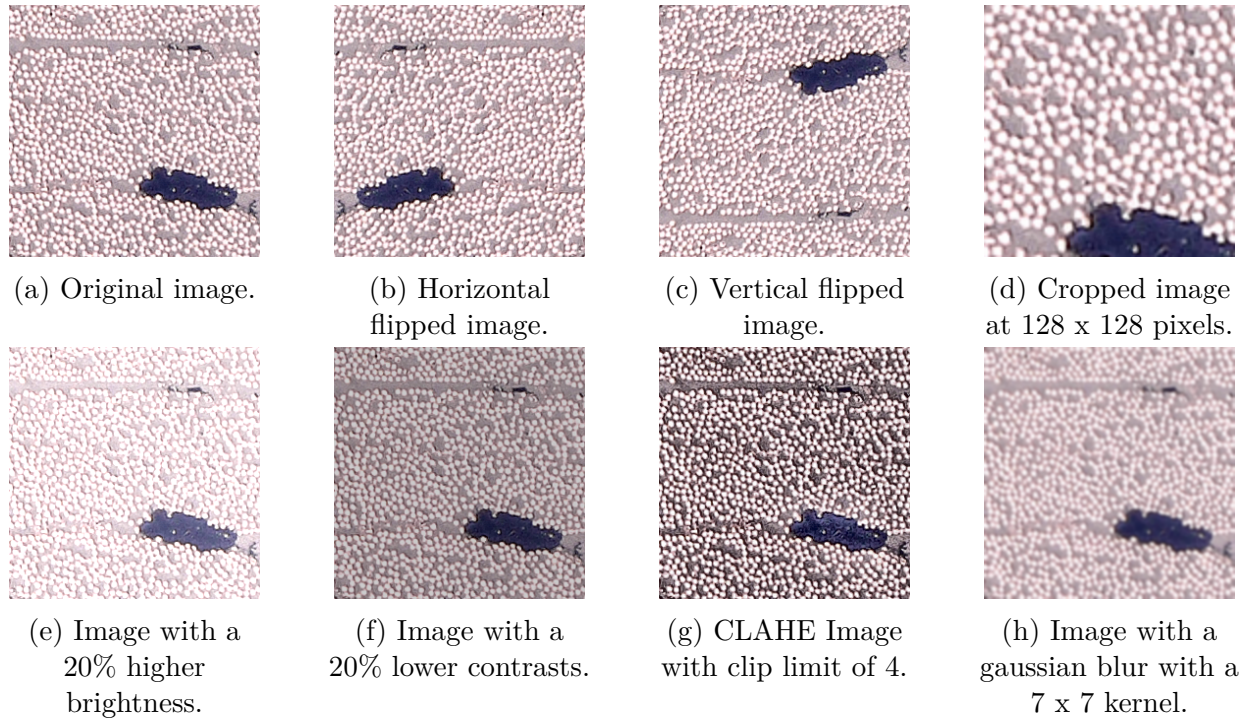


Figure 28: Example results of the seven used transformations compared to an original image.

Photometric augmentations

Second, some photometric augmentations are applied that leave the geometry of the images intact, but alter the RGB channels of the images. The brightness and contrast of the images will be randomly altered with a randomly chosen value between -20% and 20%. The contrast can also be enhanced in the images by applying Contrast Limited Adaptive Histogram Equalisation (CLAHE) to equalise the images. Adaptive histogram equalisation methods calculate multiple histograms that correspond to different parts of an image and use them to rearrange the luminance values of the image. Each pixel is then transformed according to the cumulative distribution function (CFD) of the histogram of a square area surrounding that pixel to obtain a new equalised image. However, the contrast can be overapplied in regions of the image that are near constant. The use of CLAHE helps to reduce this, as contrast amplification is limited by the use of a clip limit [39]. In the histograms the number of pixels is counted for each pixel intensity, this method then loops through all intensities with indices from zero to 255 to check if their occurrence is larger than the clip limit. For example, a clip limit of four is selected, and the count of pixels with an intensity of zero is 11. This is greater than four; thus, the count of pixels with intensity zero is clipped at four and the excess pixels counts are stored. This means that, in the end, not a single intensity from zero to 255 has a pixel count higher than the clip limit. Then the excess pixel counts are evenly distributed across all bins in the histogram of the image, as shown in Figure 29. For example, every pixel intensity gets 20 counts extra; then the maximum difference in pixel counts between different intensities after clipping and redistributing is four, which means that the maximum slope of the CDF is also four. This CDF is used to transform the pixels in the initial image into a new, equalised image. The clip limit used for data augmentation ranges from one to four, while the tile size is 8 x 8. This means that the square area around each pixel on which

a histogram is calculated is eight by eight.

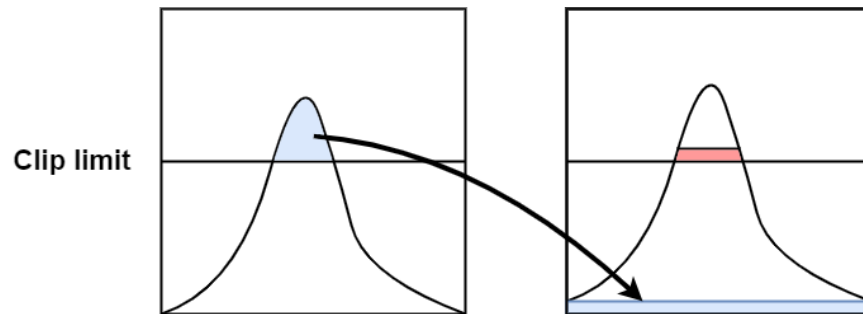


Figure 29: A schematic representation of how pixels above the clip limit are redistributed across bins in the histogram by applying CLAHE. This redistribution will cause some bins to exceed the clip limit again (shown in red).

Blur augmentation

Finally, a Gaussian blur transformation will be applied to imitate parts of the micrograph that are not good in focus. The kernel size used for the Gaussian blur is randomly chosen between three and seven with a sigma value of zero.

4 U-net model

In this chapter, the final design of the model will be discussed, as well as the metrics used to evaluate this model.

4.1 U-net architecture

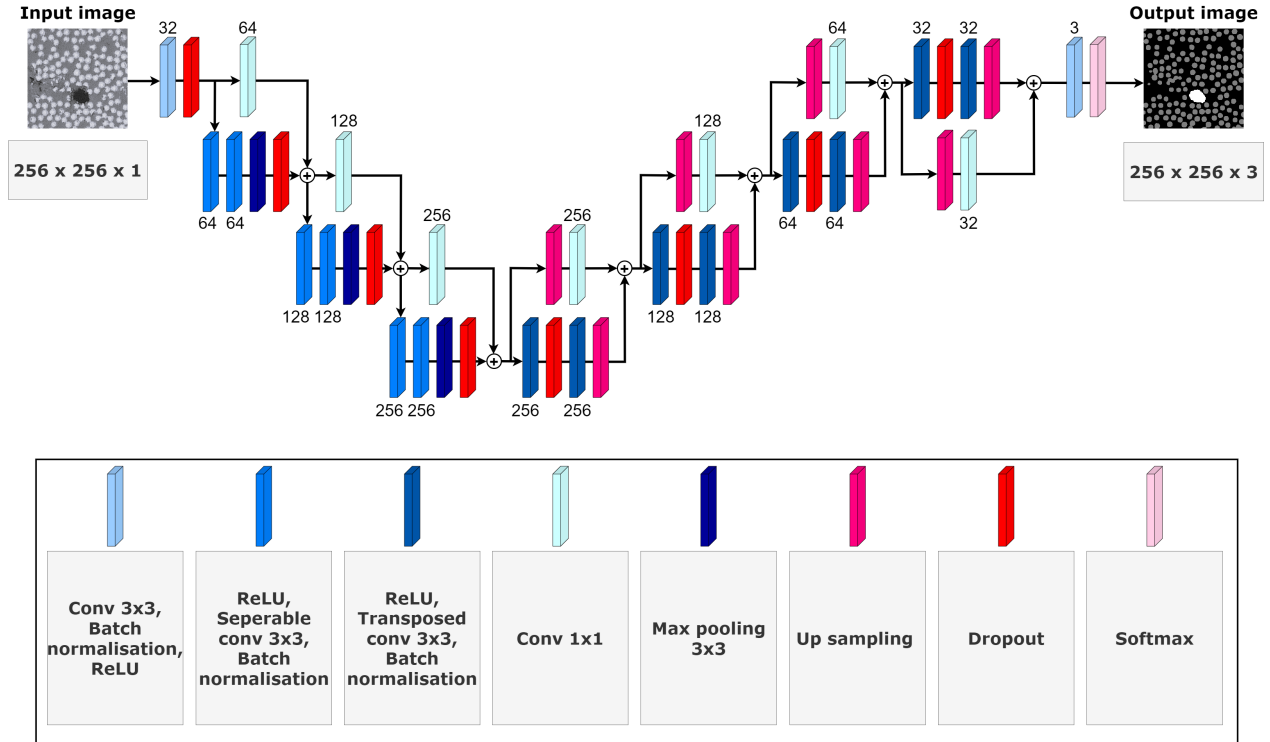


Figure 30: Architecture of the designed u-net model with depthwise seperable convolutions and residual connections. In this figure convolutional layers are indicated by “conv” and the depthwise seperable convolutional layers are called “seperable conv”. The number of filters in the relevant layers are indicated by the numbers above or below these layers.

The deep learning model in this research is based on the u-net architecture proposed by Ronneberger et al. [26] and the research on depthwise seperable convolutions by Chollet [23]. It follows the typical u-net encoder-decoder architecture, as can be seen in Figure 30, which shows the detailed network. The encoder part follows a contracting path, where the input image is reduced in size and the feature channels are increased. This is done by first using a convolutional layer with a 3 x 3 kernel and 32 filters followed by a batch normalisation layer, a ReLU non-linear activation function, and a dropout layer. This is followed by a new module consisting of a ReLU layer, a separable convolutional layer with a 3 x 3 kernel and 64 filters, which uses L2 regularisation with a regularisation parameter of 0.01, and a batch normalisation layer. This module has a linear residual connection around it that consists of a convolutional layer with a 1 x 1 kernel and again 64 filters. Then these two paths are added together. This process is repeated twice more, with the filter size increasing to 128 and then

256 for each step. The decoder part of the model starts with a ReLU layer, followed by a transposed convolutional layer with a 3 x 3 kernel and 256 filters. Then a batch normalisation layer is used followed by an up-sampling layer. The residual connection consists of an up-sampling layer and a convolutional layer with a 1 x 1 kernel and 256 filters. The two paths are combined, and this module is repeated three more times, each time with a decreasing number of filters, from 128 to 64 and finally to 32. The final layer is a 3 x 3 convolutional layer with a softmax activation function, which gives the probability distribution over all predicted classes.

The original u-net model, as shown in Chapter 2.7 Figure 9, is modified to increase the performance of the model and reduce training time. Depthwise separable convolutional modules with linear residual connections around them, based on the Xception architecture of Chollet [23], are used to reduce the number of parameters of the model. The current model as shown in Figure 30 has a total of 2058403 parameters of which 3776 are non-trainable, while the original model has more than 15 times as many trainable parameters. The reduction in the number of kernels in the different convolutional layers compared to the original number also decreased the number of parameters. Furthermore, it is chosen to transform the images to greyscale, which has only one channel instead of the red, green, and blue channels in coloured images. It is assumed that the colours in the image are not significant for the outcomes, so they are disregarded to reduce the number of parameters in the model and increase generalisation capability.

4.2 Training, validating and testing

The deep learning network will be trained with a training data set to fit the parameters of the model. Successively, the fitted model is used to validate the performance and tune the hyperparameters by using it to make predictions on observations in a second data set; the validation data set. This data is seen by the model, but the model will never learn from this data. The three data sets created in Chapter 3.4 will be used to create the training and validation data set. From each shuffled data set 80% of the images will be used for training and 20% for validation. The three data sets will be used to train three different models, each of which is trained with a different data set. The fourth data set, the test set, created in Chapter 3.4 will be used to test the fully trained model on new data from different micrographs.

During training the model is optimised with the Adam optimiser with a learning rate of 0.001. This optimiser uses a stochastic gradient descent method based on the adaptive estimation of first-order and second-order moments. This optimiser was selected for its computational effectiveness and its ability to handle large data sets and a large number of parameters [40]. The loss function used during training is the categorical cross-entropy, which will be explained in the next chapter, Chapter 4.3. Furthermore, a batch size of 16 is used and the models are trained for a total of 500 epochs. The models were implemented with Keras using Tensorflow and trained on a high-performance computing cluster that consists of one head node and eight compute nodes. Each node has 32 processor cores, these processors are Intel(R) Xeon(R) Silver 4216 CPUs. Note that this cluster is not optimised for training neural

networks as it does not contain a GPU and, therefore, the focus of this research is not on optimising the training time.

4.3 Categorical cross-entropy

The u-net model will predict to which class a pixel in the image belongs. To evaluate how good this prediction is, loss functions are used. Loss functions should return higher values when predictions are reasonable and lower values when bad predictions are made. These functions are used to optimise the model during training, as explained in Chapter 2.3.1. When the loss is lower, the model should perform better. Cross-entropy loss L_{CE} is a widely used loss function for segmentation problems [18] and is given in Equation 9.

$$L_{CE} = - \sum_{i=1}^n t_i \log(p_i). \quad (9)$$

Here t_i is the target value and p_i is the predicted value for the i^{th} class, while n indicates the total number of classes. Thus this loss function compares each probability of the predicted class to the actual class and penalises the probability of the predicted class by how far it is from the expected value corresponding to the actual class. This penalty is logarithmic; thus assigning a higher value for larger differences approaching one, while assigning a lower value to smaller differences close to zero. Therefore, the cross-entropy loss is also known as logarithmic loss, and the closer the cross-entropy loss is to zero, the better the model.

For example, a pixel belonging to a fibre in the ground truth mask is predicted to have a probability of 0.7 being a fibre, 0.2 belonging to the matrix, and 0.1 being a void. Thus, this pixel is predicted as a fibre, since this class has the highest probability. The loss is then calculated with $t = [0, 1, 0]$, since the ground truth class of the pixel is the second class (fibres), and $p = [0.2, 0.7, 0.1]$. This gives a loss value of 0.15, which is the value that will be minimised by the model. After each prediction the u-net makes, the loss is calculated for every pixel in an image. This is done for every image in the train and test set. The average loss can then be plotted over the number of epochs to observe the training progress of a model's training.

4.4 Evaluation metrics

The u-net will be evaluated during training, validation, and testing with different metrics. This chapter explains the accuracy and IoU metrics used to assess the model's performance.

4.4.1 Accuracy

The accuracy of the model will be calculated to evaluate the performance of the u-net models. Accuracy measures the frequency of predictions that match the integer labels of the ground truth masks and is calculated as shown in Equation 10.

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_{\text{correct}} + N_{\text{incorrect}}}. \quad (10)$$

Thus, the accuracy divides the number of the correctly predicted pixels, N_{correct} , by the total number of analysed pixels, $N_{\text{correct}} + N_{\text{incorrect}}$. This is a good metric to evaluate the u-net models, as higher accuracy indicates a model that is better at correctly predicting labels. However, accuracy can also be sensitive to unbalanced data sets where one class is more representative than other classes. This is shown in Figure 31. Here, a ground truth mask is shown, with a white fibre located just off the centre. Below this ground truth mask, four different predictions are shown. Prediction 1 perfectly matches the ground truth mask. Then, in prediction 2, the fibre location is slightly shifted. In prediction 3 this is shifted even further down. And prediction 4 shows a fibre completely predicted in a different corner. Below the predictions, a visualisation is shown of what a subtraction of the predicted masks from the ground truth mask would look like. Here, black pixels indicate correctly predicted pixels (TP), and red pixels indicate pixels that should belong to fibres but are incorrectly predicted (FN). Lastly, blue pixels indicate pixels that are predicted to belong to fibres but are not indicated as fibres in the ground truth image (FP). The calculated accuracy of each of the four predictions is shown at the bottom of the figure. As can be seen, prediction 1 has an accuracy of 1.00. However, prediction 4 still has an accuracy of 0.95, whereas the prediction of the location of the fibre is completely off. This illustrates that, in this case, accuracy is not a good measure because not all classes are evenly represented in this example. This is also the case for FRP micrographs, as the void volume fraction is often below 5%.

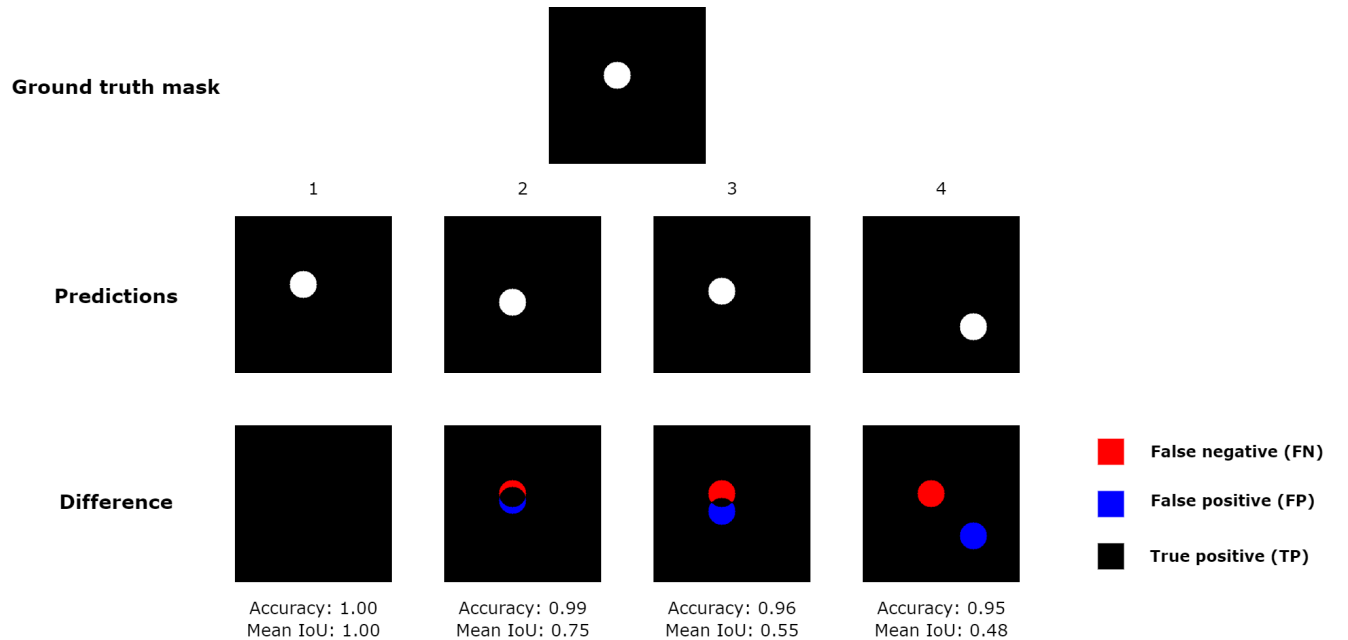


Figure 31: Accuracy and mean IoU calculated and compared for four different predictions of a single fibre.


$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}} = \frac{\text{Diagram of overlapping squares}}{\text{Diagram of union area}}$$


Figure 32: Visual representation of the Intersection over the Union calculation.

4.4.2 Intersection over Union

A metric that is less sensitive to unbalanced classes in data sets is the Intersection of the Union (IoU) metric, which is commonly used in the evaluation of image segmentation results [41]. This metric divides the overlap area by the union area, Figure 32, per class by looking at the predicted class of a pixel and the ground truth class of a pixel. Per class, the IoU can be calculated as:

$$IoU = \frac{TP}{TP + FP + FN}. \quad (11)$$

Here, the true positive (TP) pixel represents the pixels that are predicted correctly. The false positive (FP) pixels are pixels that are predicted as the relevant class but are not indicated as that class in the ground truth mask. The false negative (FN) pixels are pixels in the ground truth mask that are not predicted as that class by the algorithm. Thus, if the IoU is 1, then the prediction is completely correct, and the worse the prediction, the lower the IoU. For multiclass predictions, the IoU is calculated for every class and then the mean of these values is calculated to get a mean IoU for one prediction. This is illustrated in Figure 31. Here, prediction 1 is completely correct, which gives an IoU of 1. Since both the IoU of the fibres and the matrix material class are 1. Prediction 2 is almost correctly predicted and gets a mean IoU of 0.75. Then when the prediction of the fibres gets more off in prediction 3 the mean IoU drops down to 0.55. Finally, when the two fibres are not even close to each other anymore in prediction 4 the mean IoU becomes 0.48. This is because the IoU of the matrix material class is 0.95 and the IoU of the fibre class is 0, since the fibres are not predicted correctly. This gives a mean IoU of 0.48.

5 Results

In this chapter, the results of three u-net models trained with the three different augmented data sets are discussed. Table 10 shows an overview of the three different models with their corresponding accuracy and IoU during training, validation, and testing. During training, the models are saved after every ten epochs. The best-performing model of these saved models is used as a final model for testing and presented here. All models will be discussed in the rest of this chapter and compared with each other. The results of the models trained with the non-augmented data sets can be found in Appendix C.

Table 10: The three models trained with the different data sets and their accuracy and IoU results for training, validation and testing.

Model	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
Augmented data set I	0.90	0.85	0.87	0.84	0.73	0.73
Augmented data set II	0.93	0.87	0.68	0.88	0.76	0.50
Augmented data set III	0.93	0.87	0.90	0.85	0.66	0.79

5.1 Model I: Augmented data set I

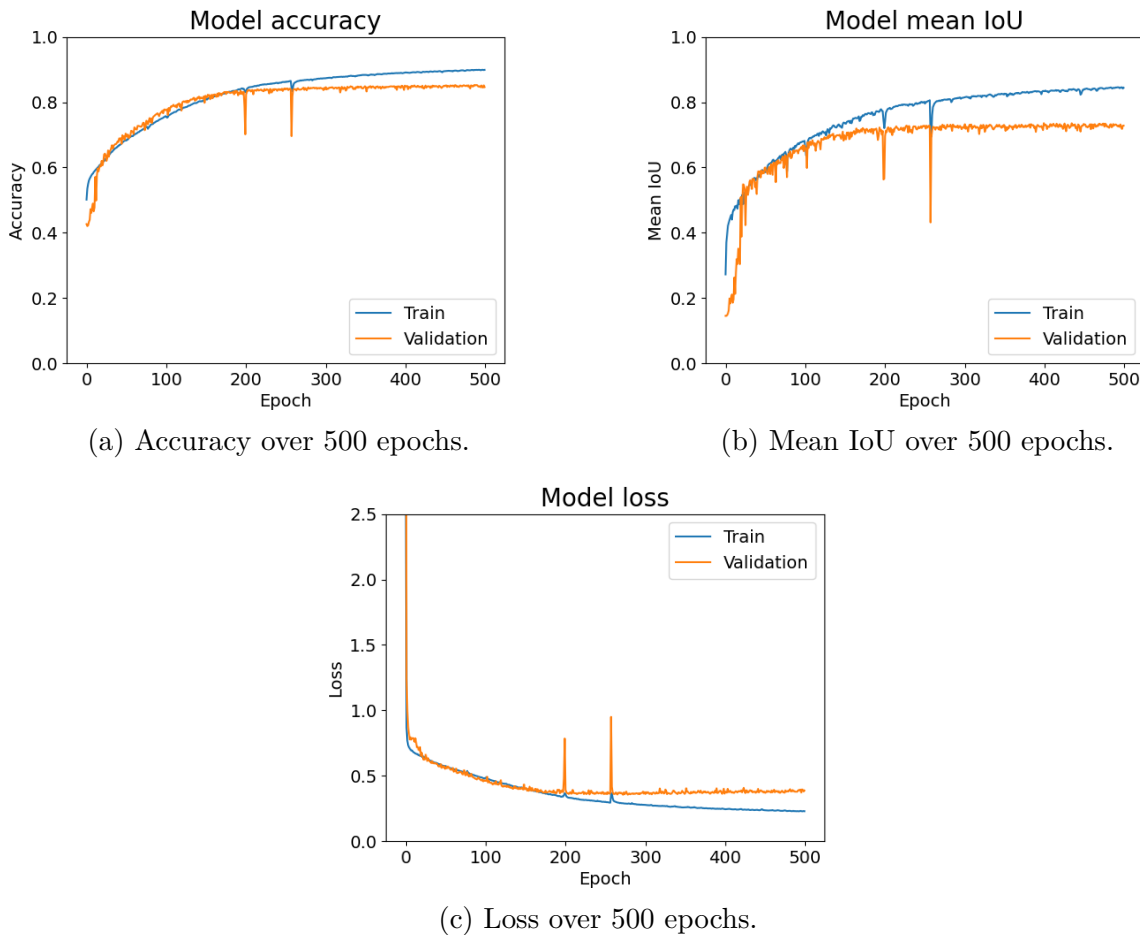


Figure 33: Plots of the u-net model trained with the augmented data set I.

Data set I is augmented using the augmentation method described in Chapter 3.5 and used to train the u-net model for 500 epochs. Figure 33 shows the accuracy, IoU and loss plots made during training the model for the train and validation data. As can be seen in the figure, the model converges after approximately 200 epochs. Figure 33c shows that the loss of training data decreases to a value of approximately 0.23, while the validation loss converges to a value of approximately 0.39. The training loss has a steeper slope and can still decrease more when the model is run for more epochs. However, this decrease in loss is mostly caused by overfitting the model. Overfitting often occurs because of a limited number of (diverse) images. The network is able to remember and learn features from the training data; however, it cannot use this knowledge on an unobserved data set [36]. After 500 epochs, the training accuracy has reached a value of 0.90 and the validation accuracy a value of 0.85, as shown in Table 10. Additionally, the training set has reached an IoU value of 0.84 and the validation set has an IoU of 0.73. These results are as expected, as the validation accuracy is typically lower than the training accuracy, since the model has not seen the validation images before. That is, the validation images are used to calculate the validation accuracy, IoU and loss in

each epoch, but they are not used to modify the model and will thus not be remembered. Thus, the model demonstrates good results on the training and validation data. However, it is of great interest if the model also performs well on a completely new micrograph. Therefore, the model will make predictions on the test data set. As shown in Table 10 the accuracy reached during the testing is 0.87 and the IoU is 0.73. These results are close to the validation results, which can be explained by the test data that contain images similar to images present in the augmented data set I. The fibres in the test data set have a fibre radius of seven pixels, while the fibres in data set I have a radius of three pixels. However, data augmentation also added images with larger fibre sizes because of the random crop method. This crops part of the image and scales it back to the image size of 256 x 256 pixels, which enlarges the fibres.

Figure 34 shows an example of a prediction of the test data set with this trained model. As can be seen, the predicted fibres are slightly smaller than the fibres in the ground truth mask. This is probably due to the data set mainly containing smaller fibre sizes than those present in the test set. Furthermore, the voids in the images are overdetected; predicted to be larger than it actually is. All of the test images demonstrate this, and even two additional voids are predicted, which are not present in the input images. This is most likely because the images in the data set contain more and sometimes larger voids. Thus, the void predictions need to be improved in this model. Note that in images that contain voids the fibres are indicated by a grey colour, whereas the voids are white and the matrix material is black. However, if no voids are present, the fibres are white and the matrix material black. The colours in Figure 34d is used to illustrate the difference between the prediction and the ground truth image, these images are used throughout this thesis. Here, red pixels indicate pixels that are predicted as matrix material but belong to fibres in the ground truth mask. The blue pixels are pixels that are predicted as fibres but are actually matrix material. Yellow pixels indicate pixels that are predicted as voids, but are matrix material in the ground truth mask. Pink pixels are the opposite; pixels that are predicted as matrix material, but belong to voids in the ground truth masks. However, in this example, no pink pixels are present.

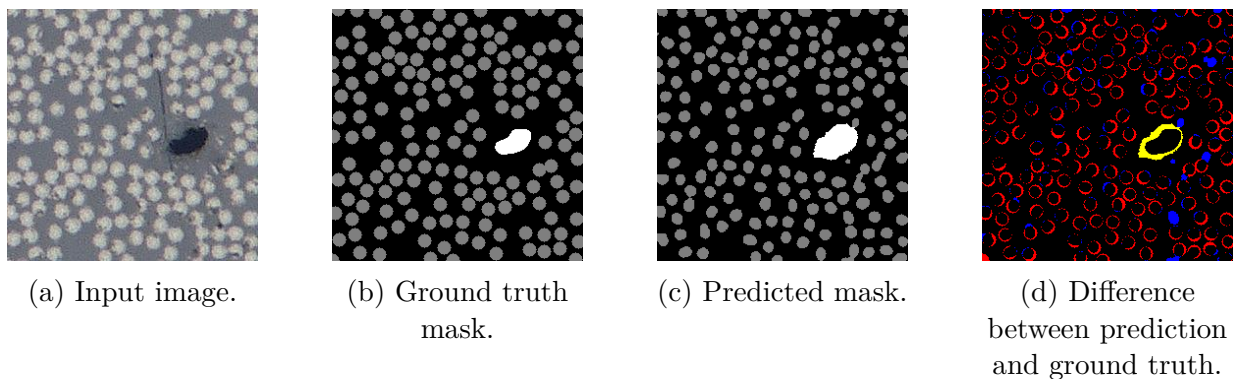


Figure 34: Prediction of the trained u-net model with data set I augmented for a test image with an accuracy of 0.88 and an IoU of 0.70.

5.2 Model II: Augmented data set II

Another u-net model is trained with the augmented data set II, containing images with two different sizes of fibres. Figure 35 shows that this model converges after approximately 200 epochs. The accuracy reached with the train data is 0.93 and the IoU is 0.88. With the validation data, an accuracy of 0.87 is reached, while the IoU reaches a value of 0.76, which is also summarised in Table 10. Furthermore, the loss decreases to a value of 0.16 for the training data and 0.33 for the validation data.

These metrics are slightly better than the model trained with the augmented data set I, but without significant improvement. However, the predictions of this model on the test set are worse than those of the previous model trained with the augmented data set I. The test accuracy is 0.68 and the test Iou is 0.50. Figure 36 shows a poor prediction of this trained model. As can be seen, the fibre sizes in the predicted mask are larger than those in the ground truth mask. This is also seen in the test results of a model trained with the non-augmented data set II, as outlined in Appendix C.2. This is likely because there are very few training images in the augmented data set II that contain fibres with a radius of seven pixels, which is the fibre size present in the test data set. Half of the images present in data set II namely have a fibre radius of three pixels and the other half have a fibre radius of 10 pixels. The data augmentation method is used to bring more variety in these fibre sizes. However, the fibre radius can only be increased by random cropping in the data augmentation process. This cropped part of the image is then scaled back to the image size of 256 x 256 pixels, which enlarges the fibres in the image. Therefore, at least half of the training images have a fibre radius larger than 7 pixels. The other half of the image can experience a random enlargement of the fibres; however, this means that fewer images will contain fibres with a radius of seven pixels than in the augmented data set I, where this was possible for all 500 images. Therefore, the model trained with the augmented images from data set II performs worse on the test data than the model trained with the augmented images from data set I. Furthermore, because all images with a fibre radius of 10 pixels in the II data set do not contain any voids, this model also has more difficulty correctly predicting voids compared to the model trained with the augmented data set I, which contains 405 images with voids. Figure 36 shows such an underpredicted void.

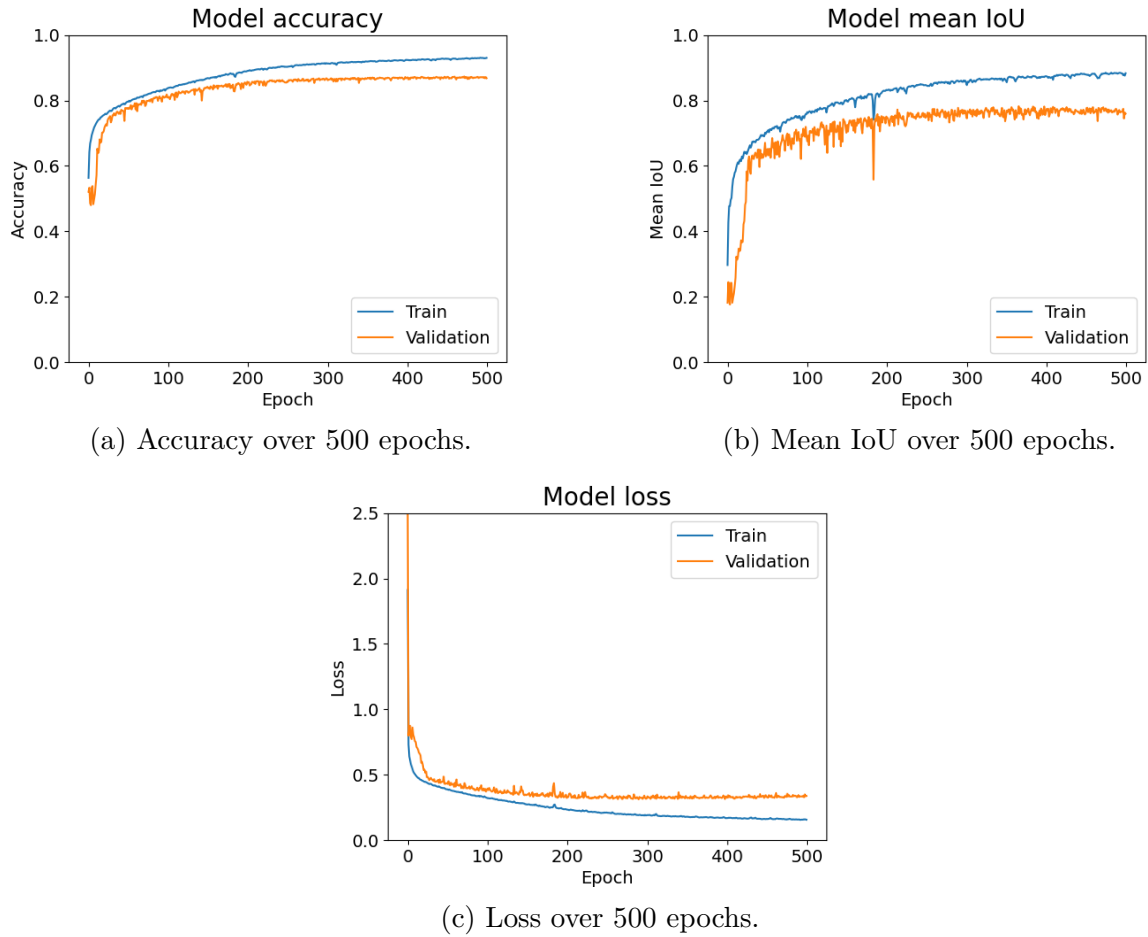


Figure 35: Plots of the u-net model trained with the augmented data set II.

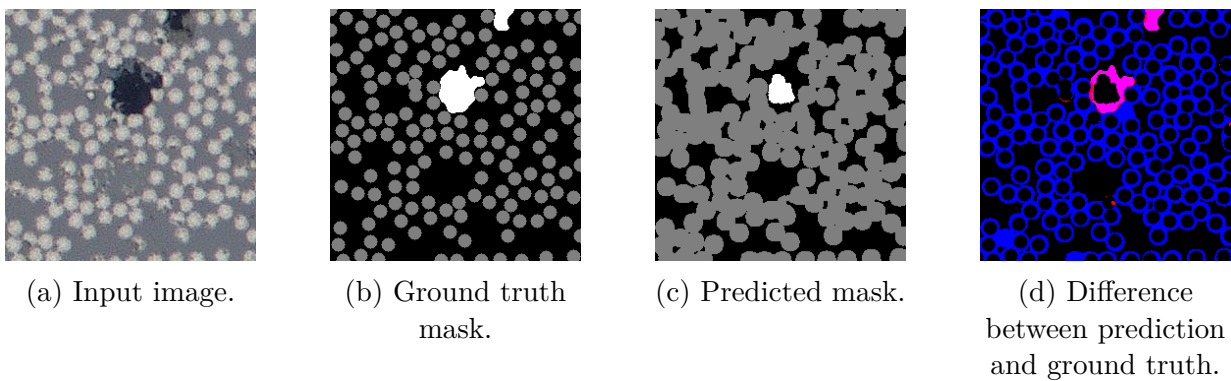


Figure 36: Prediction of the trained u-net model with augmented data set II for a test image with an accuracy of 0.67 and an IoU of 0.35.

5.3 Model III: Augmented data set III

A third model is trained with augmented data set III. Figure 37 shows the evaluation plots of the u-net model trained with these augmented images for 500 epochs. As can be seen, the model converges after approximately 100 epochs. As can be seen in Figure 37a the training accuracy increases to a value of approximately 0.93 after 500 epochs and the validation accuracy converges to 0.87. The IoU during training increases to a value of 0.85, while it converges to 0.66 during validation. Lastly, the loss during training drops to a value of 0.15 while during validation it only reaches 0.36. The model performs approximately the same on the train and validation data, except that the validation IoU is significantly lower than the previous two models. This is because the data in data set III contains more variety, making it harder for the model to make a good prediction.

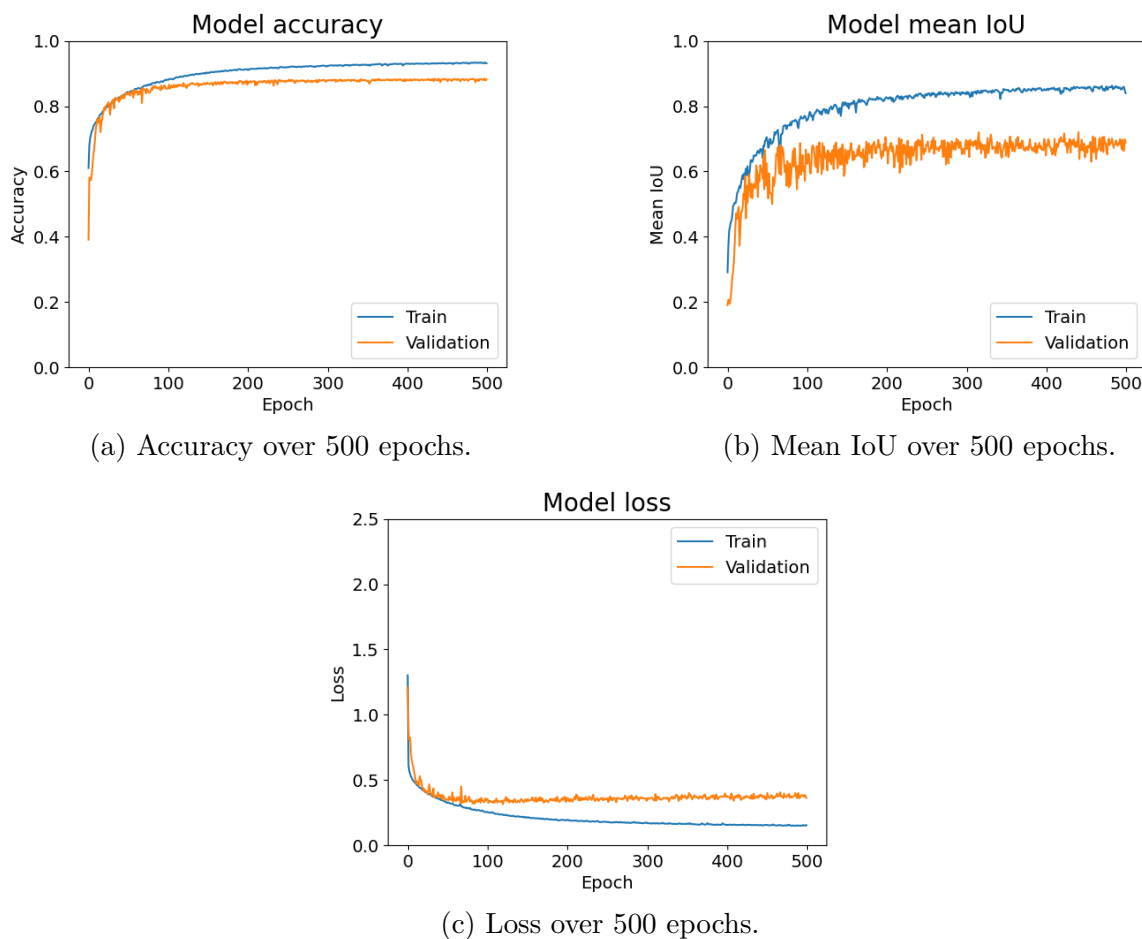


Figure 37: Plots of the u-net model trained with augmented data set III.

This trained model III is used to predict the segmented masks of the 32 images in the test data set. The average accuracy achieved is 0.90, and the mean IoU is 0.79, as shown in Table 10. This is significantly better than models I and II. This shows that having a more diverse data set containing images of multiple micrographs does increase the performance on the test data set. However, it should be noted that the test data set is relatively small due

to a lack of available data. It is not guaranteed that the model will produce the same results when applied to data other than the test set. Figure 38 shows the image that achieved the best IoU score by testing the model. The void is well predicted and the fibres are also mostly in accordance with the fibres in the input image. Figure 39 shows the image that scored the worst during the testing of this model. As can be seen, the fibre prediction is still pretty accurate, but because the void is not recognised at all, the IoU is only 0.54. A reason for this can be the small amount of training data that contains voids, as only 19.1% of the images in data set III contain voids. Furthermore, there are a few fibres predicted completely wrong, indicated by clear blue spots that are the size of a fibre in Figure 39d. However, this does not always indicate that the model does not perform well. For example, when zoomed in on the two blue fibres in the top right corner of the image, as shown in Figure 40, it can be seen that the model's prediction can be better than the ground truth mask. The two encircled fibres visible in the input image, Figure 40a, are not in the ground truth mask, Figure 40b, but are predicted by the model in Figure 40c. The average predicted fibre volume fraction is 0.36 and the average void volume fraction is 0.002, while these values of the ground truth masks are respectively 0.37 and 0.003. This again demonstrates that the fibres and voids are slightly underestimated, but that the model is able to make accurate predictions that can approximate the fibre and void volume fractions.

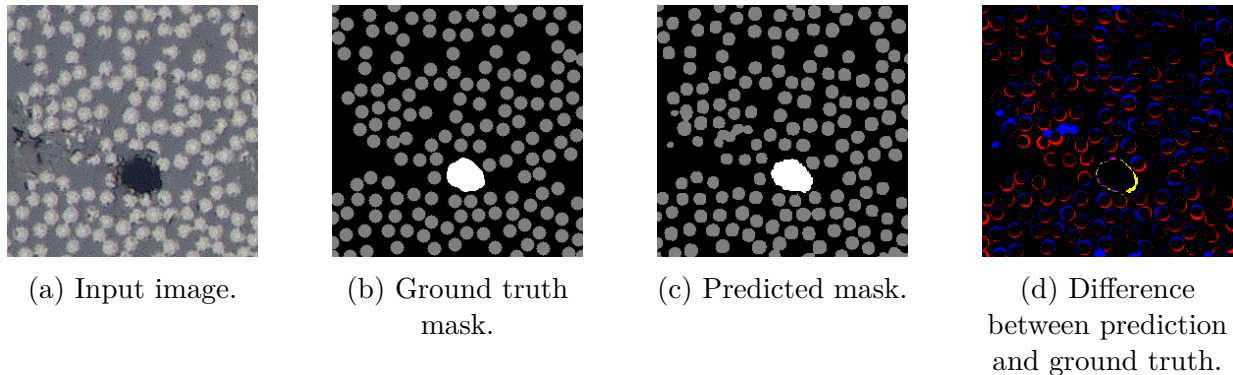


Figure 38: Prediction of the u-net model trained with augmented data set III for an example image with the best achieved Iou of 0.84 and an accuracy of 0.91.

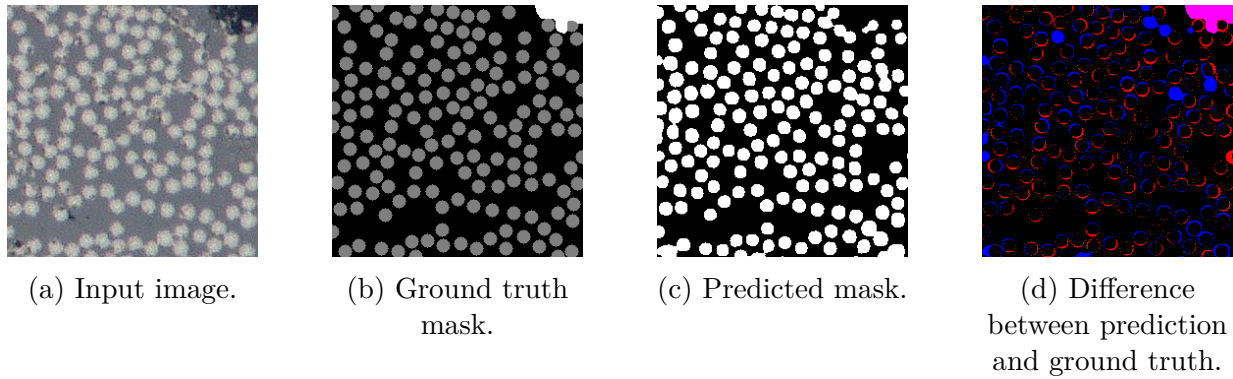


Figure 39: Prediction of the u-net model trained with augmented data set III for an example image with the worst achieved IoU of 0.54 and an accuracy of 0.89.

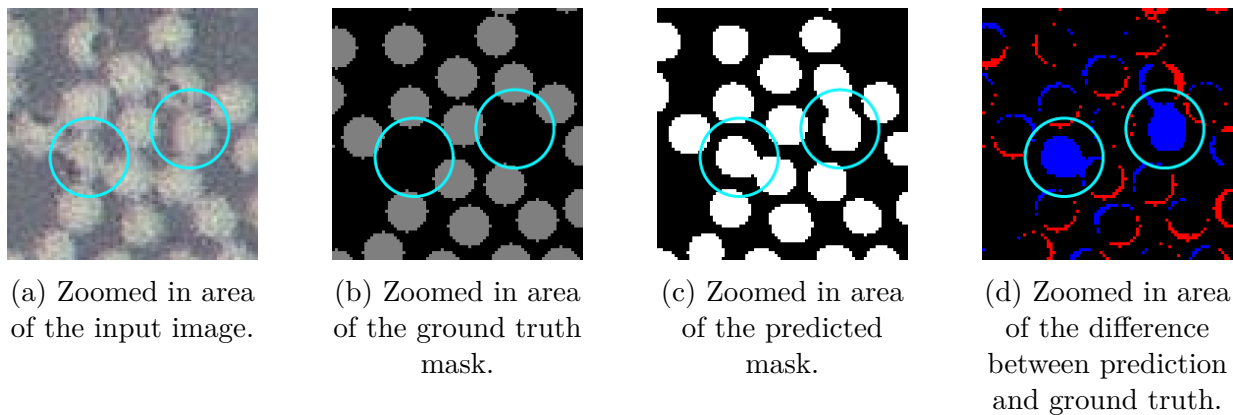


Figure 40: Zoomed in images of 85 x 85 pixels of the same area of the images shown in Figure 39. These images show that the prediction of the model can be better than the ground truth mask, as the two encircled fibres are not present in the input image.

5.4 Comparison of the three models



(a) Accuracy results of the three different models.

(b) IoU results of the three different models.

Figure 41: Bar graphs comparing the training, validation and test results of Model I, II and III.

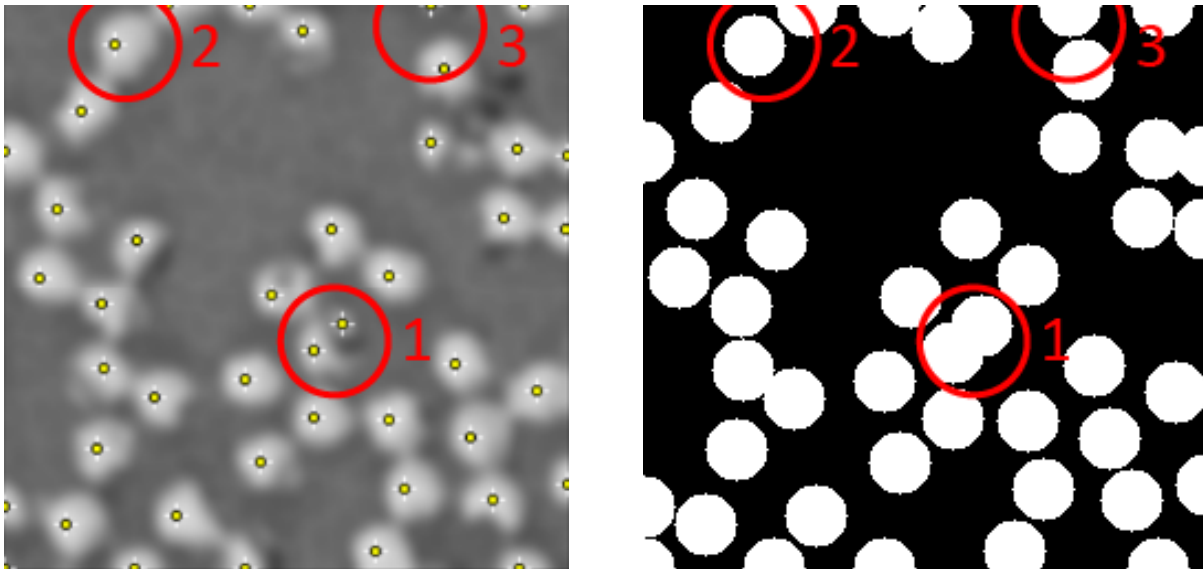
Figure 41 compares the training, validation, and test results of the three models. When looking at the accuracy results, it is clearly visible that model III scores best on all three phases. For the IoU results model II scores best in the training and validation phase, however, it scores worst in the test phase. After all, the test phase is the most important, as this indicates how well the model performs on new data. The high results during training and validation can be caused by overfitting. Therefore, model III has the best overall performance as expected, as it is trained with the data set that has the highest diversity. Thus, this model will be the final outcome of this research. Some feature maps of this model are visualised in Appendix F to see what happens inside the model; these figures are enlarged versions of the feature maps shown in Figure 10. Furthermore, to determine whether this model can correctly predict an image with a single fibre when trained with images containing multiple fibres, an additional test was performed and the results are presented in Appendix E. In Chapter 6.2, the effects of various parameters on this model will be demonstrated to justify the decisions taken to design this model.

6 Discussion

In this chapter the mask generation procedure is discussed as well as a justification of how parameters for the final model were chosen.

6.1 Mask generation

Masks were required from the input images to train and test the model. The masks need to indicate which pixels belong to fibres, matrix material, and voids. The semi-automatic process used for this, as described in Chapter 3.3, is less accurate than labelling pixels by hand due to various reasons. Figures 42 and 43 illustrate errors made in the mask caused by the mask generation process.



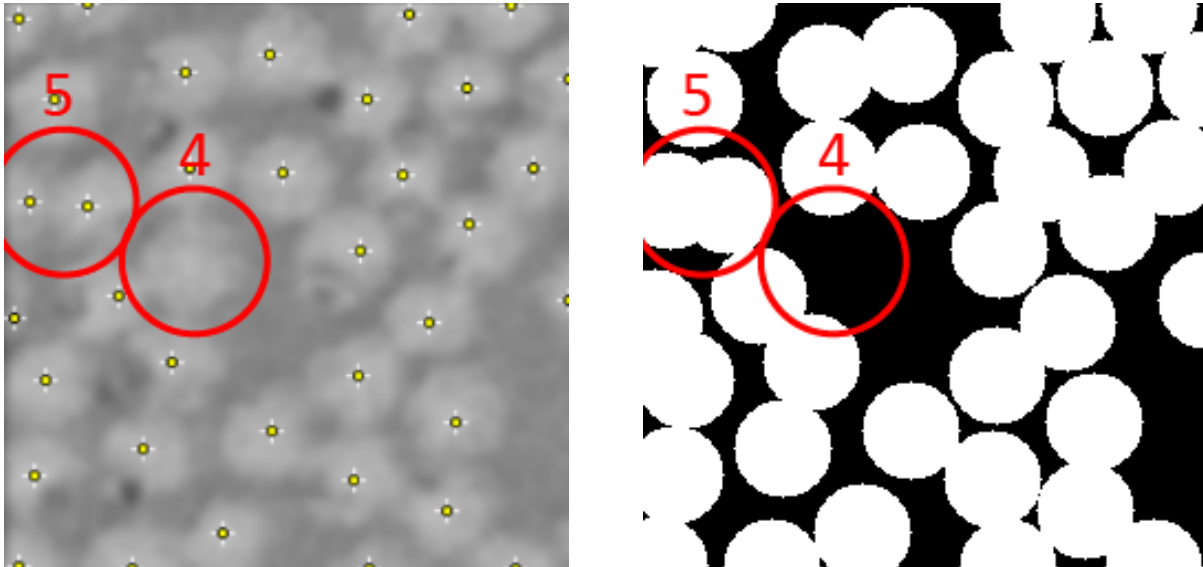
(a) Image after preprocessing steps in ImageJ with the local maxima indicated by white crosses with yellow circles in it.

(b) Corresponding mask made by using the local maxima as fibre centres and plotting circles with a radius of 14 pixels around them.

Figure 42: Example of an image and corresponding mask that show imperfections.

First, the “Find maxima” function in ImageJ indicates the local maxima in the pixel intensity in an image. In some locations, the local maxima do not belong to the fibres, and, in general, these local maxima do not precisely describe the centres of the fibres. This means that fibres can appear in the masks at places where they are not present in the original image and that many fibres are shifted a bit in the mask because the predicted centres are off. As can be seen in Figure 42, circle number 1 shows an imperfect fibre that resulted in two local maxima; therefore, two fibres were plotted in the mask instead of one. During the polishing process of the specimens, some fibre ends can become damaged and are therefore not visible as circles. The mask generation process currently is unable to handle the recognition of the broken edges of fibres or imperfect fibres with the chosen parameters and method. Circle 2 shows a fibre of which the local maximum does not overlap the centre of the fibre, making

the fibre in the mask shift too far to the left.



(a) Image after preprocessing steps in ImageJ with the local maxima indicated by white crosses with yellow circles in it.

(b) Corresponding mask made by using the local maxima as fibre centres and plotting circles with a radius of 14 pixels around them.

Figure 43: Example of an image and corresponding mask that show imperfections.

The opposite also occurs, a fibre in the original image does not have a local maxima with the chosen parameters. Therefore, when the centre is not recognised, no fibre will appear in the corresponding mask, as shown by circle 4 in Figure 43. However, this is observed to occur less often than to predict more fibres that are actually present. Another problem that this mask generation process cannot handle well is the recognition of fibres that are partially present at the edges of an image, which arises as a result of cropping the micrographs. When only a part of the fibre is shown in the original image, a local maximum is often recognised, but this maximum mostly does not correspond to the fibre centre. For example, when only one third of the fibre is present in the original image, the local maximum will not accurately pinpoint the centre of the fibre, as it is not even in the image. Then, in the mask generation process, a circle is drawn around this maximum with a fixed radius, resulting in a larger area being assigned to the fibre than is necessary, as shown by circle 3 in Figure 42. Here, half a fibre shows up in the mask, whereas this fibre is barely visible in the original image. Therefore, in the majority of cases more pixels are classified as fibre than is actually the case.

In Figure 43, some fibres are cut off at the edges, making them appear larger in the mask than their actual size. Furthermore, this leads to fibres overlapping in the mask, which is not possible in reality. An example of this is shown by the fifth red circle in Figure 43. This overlap also often occurs when the local maxima are not located in the centres of the maxima.

Lastly, in the mask generation process it is assumed that the fibres are perfectly circular and all have the same fibre size when looking at their cross-section. When looking at the

training images, it is clear that this is not the case. This is because the fibres do not always lay completely straight in one direction, the fibres might be a bit curved or the cross-section is not made perpendicular to the fibres, which can make their cross-section more ellipse shaped. Moreover, depending on the manufacturing process, the cross-sections of carbon fibres do not always turn out perfectly circular. In addition, the ends of the fibres can also be chipped off at the sides during the polishing process before making a microscopy image. However, these defects also influence the fibre volume fraction in an unfavourable way. The fibre volume fraction in the mask will become higher or lower than the actual fibre volume fraction. Therefore, it is not expected that these generalisations that do not include these defects in the masks would have a significant negative influence on the results in the fibre volume fractions.

6.1.1 Fibre and void volume fraction

The fibre and void volume fraction can be calculated directly from the ground truth masks and the predicted masks from the models later on. When looking at the fibre volume fractions, it is noticed that these are relatively low. The fibre volume fraction of the test set is 0.37 and the void volume fraction is 0.003, while the fibre volume fraction is expected to be above 0.5. However, when looking at the masks they correspond relatively well to the corresponding images. It is noted that increasing the fibre radius by one pixel significantly increases the fibre volume fraction. When the fibre radius of the fibres in the test set is increased from seven to eight, the fibre volume fraction goes from 0.37 to 0.48. Thus, this method of using perfect circles with a fixed diameter is quite sensitive to small changes.

6.2 Effect of various parameters on model III

To develop the best performing model, model III from Chapter 5.3, several parameters, such as data augmentation, amount of data, learning rate and batch size, were studied. It was observed that the performance of the model improved when the parameters were adjusted accordingly. This sub-chapter discusses the influence of some of these parameters on the results. The different models resulting from the use of different parameters will be compared with the best-performing model III as described in Chapter 5.3. This model has also been compared with the original u-net model proposed by Ronneberger et al. [26], the results of which can be found in Appendix D.

6.2.1 Data augmentation

Data augmentation is used to add variability in the training images in all three models from Chapter 5. To investigate the influence of the data augmentation method used, another model will be trained with the same parameters but with the non-augmented data set III. Table 11 shows the results of training, validating, and testing this model and compares it with the model that was trained with augmented data. The model with non-augmented data scores higher on the tracking and validation data, but significantly lower on the test data. This is caused by the difference in variety in the train data. The non-augmented data appears very similar, because the 256 x 256 images come from only a few larger micrographs. Thus,

the model performs well on these images, since it has seen many similar looking examples. However, the model knows the training data and noise too well, which is a sign of overfitting, as it cannot make a good prediction on the new data from the test set.

Table 11: The accuracy and IoU results of a model trained with a non-augmented version of data set III and the original augmented version.

Model	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
Non-augmented data set III	0.95	0.92	0.65	0.87	0.78	0.48
Augmented data set III	0.93	0.87	0.90	0.85	0.66	0.79

Figure 44 shows a prediction of the model trained with non-augmented data on the test set. Most of the fibres in the prediction, Figure 44c, are predicted to be larger than they actually are. This is also seen in the prediction of the model trained with data set II in Chapter 5.2. Therefore, the augmentation method does add the variety to the data that is needed to make more general predictions on different RFP micrographs.

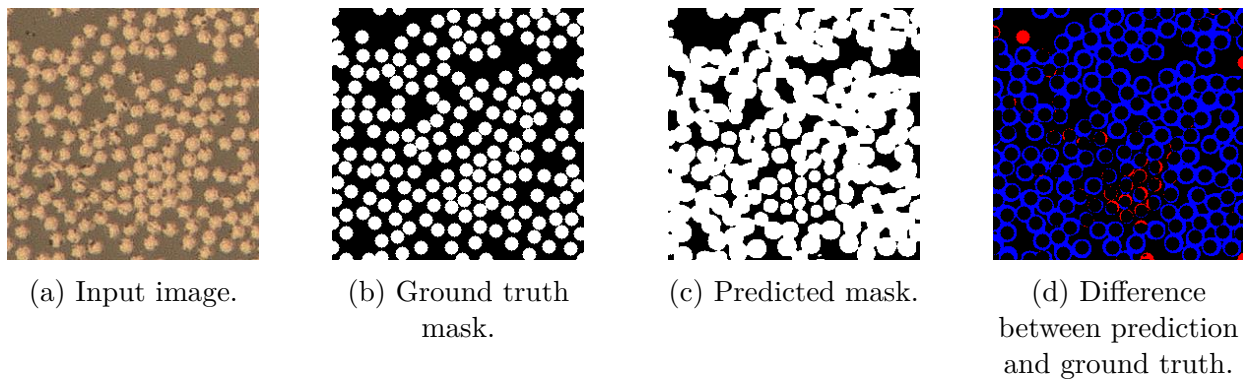


Figure 44: Prediction of the u-net model trained with the non-augmented data set III for an example image with an accuracy of 0.69 and an IoU of 0.53.

6.2.2 Amount of data

In generally deep learning models need a lot of training data, but u-net models are known to require very few augmented annotated images during training [26]. Therefore, it is interesting to see how a model trained with fewer images than the 1000 that are included in the augmented data set III performs. For this, the data set is reduced to 100 augmented images by taking 10 images from each set of images with a different fibre size, as shown in Figure 26. Furthermore, data augmentation makes it relatively easy to increase the size of a data set by adding additional augmented images. This is done with data set III; every image is now augmented twice, which increases this data set to 2000 images. Both trained models are compared with the final model from Chapter 5.3. Table 12 shows the accuracy and IoU results of all tree models. As expected, the model trained with only 100 images performs worse than the other two models, because it has seen fewer data to learn from. However, the model trained with 2000 images does not perform significantly better than the model with

1000 images. This is most likely because data set III contains 10 batches of 100 images that look similar. The data augmentation method adds variability to these images to some extent. However, when more of these images are added, the images again begin to look similar. Another reason could be that the limit of the optimal result has been reached. The ground truth masks are not perfect, as explained in Chapter 6.1, thus, the predictions can also never be perfect.

Table 12: The accuracy and IoU results of a model trained with 100 and 2000 augmented images of data set III compared to the original model trained with 1000 augmented images.

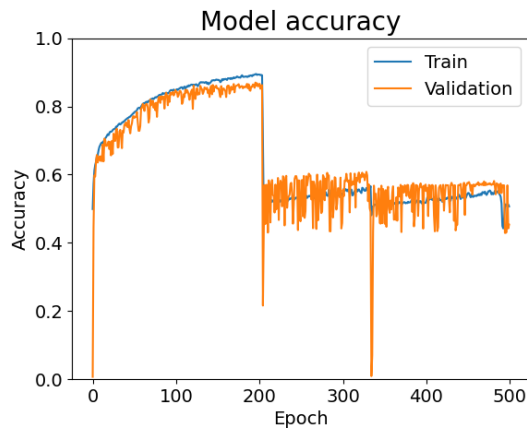
Model	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
Augmented data set III 100 images	0.89	0.79	0.50	0.81	0.83	0.66
Augmented data set III 1000 images	0.93	0.87	0.90	0.85	0.66	0.79
Augmented data set III 2000 images	0.93	0.90	0.90	0.85	0.70	0.76

6.2.3 Model hyperparameters

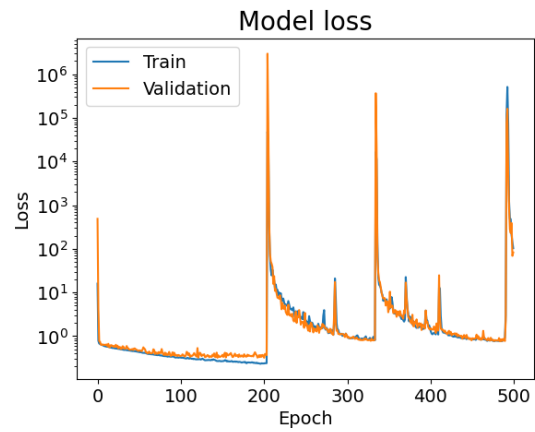
The models contains several hyperparameters that can be tuned to get better results or speed up the training process. The learning rate and batch size are two of these hyperparameters that will be discussed here. The final model uses a learning rate of 0.001, but values of 0.01 and 0.0001 were also tested, as shown in Table 13. The results in accuracy and IoU values are not significantly different, making all learning rates tested usable. However, the training process is highly influenced by this parameter and, for effective deep neural network training, the learning rate is an import parameter to tune [42, 43]. Large learning rates can skip the optimal solution and result in unstable training. This happened with the model that used a learning rate of 0.01. Figure 45 shows the accuracy and loss plots during training. After 200 epochs the model reaches its highest accuracy and lowest lost values, and the saved model at this time is used for the results shown in Table 13. However, after 200 epochs the accuracy suddenly goes down again and does not increase any more for the remaining 300 epochs. This is a typical behaviour that is seen in models with a too high learning rate. The weights change a lot due to this learning rate and are moving away from the global minimum, which results in a lower accuracy. The model can even get stuck at a local minimum and fails to improve, which is what happened in this case after 200 epochs. Too small a learning rate can lead to slow convergence and longer training times [42]. Figure 46 shows the accuracy and loss plots of the trained model with a learning rate of 0.0001. This model indeed converges slower than the model with learning rate 0.001 from Chapter 5.3. Since the accuracy continues to improve around 500 epochs in Figure 46a. Therefore, the final model uses a learning rate of 0.001, as this gave the best training results.

Table 13: Comparison of models trained with the augmented data set III with different learning rates.

Learning rate	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
0.01	0.89	0.87	0.88	0.78	0.60	0.75
0.001	0.93	0.85	0.90	0.89	0.66	0.79
0.0001	0.89	0.86	0.87	0.81	0.65	0.73

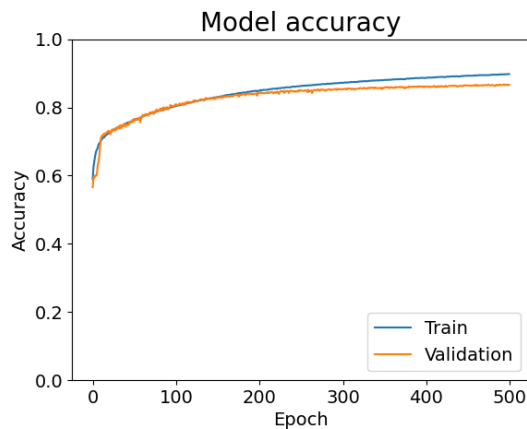


(a) Accuracy over 500 epochs.

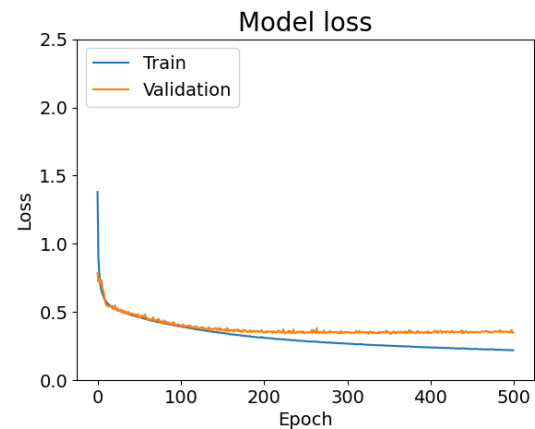


(b) Loss over 500 epochs.

Figure 45: Plots of the u-net model trained with augmented data set III with a learning rate of 0.01, which show an unstable training process due to a too high learning rate.



(a) Accuracy over 500 epochs.



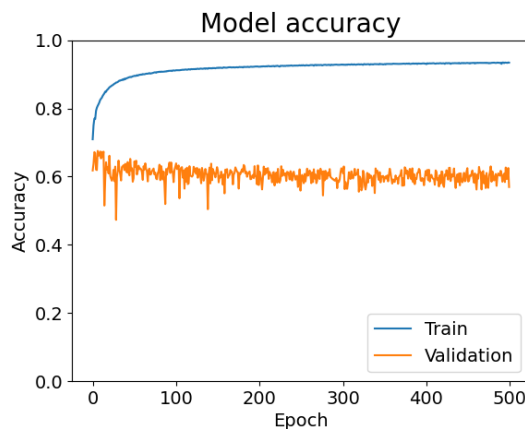
(b) Loss over 500 epochs.

Figure 46: Plots of the u-net model trained with augmented data set III with a learning rate of 0.0001, which shows a slower convergence due to a too low learning rate.

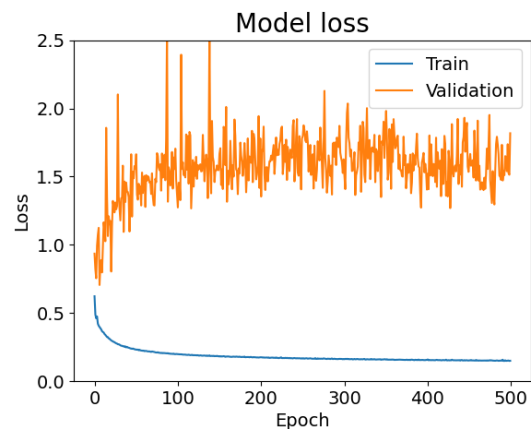
Secondly, the batch size is adjusted. The original u-net model proposed by Ronneberger et al. [26] uses a small batch size of one. Small batch sizes can be computationally expensive and time consuming, but achieve high accuracy. However, small batch sizes can also be more susceptible to random fluctuations in the training data. This is also observed when using a batch size of one with the model trained with the augmented data set III, as shown in Table 14 and Figure 47. The validation and test results are significantly lower than the higher batch sizes of eight, 16 and 32. The accuracy and loss graphs demonstrate that the performance begins to decline after 20 epochs and fluctuates heavily with each epoch. The gap between the training and validation data results is significant and shows that the model is not resistant to the fluctuations in the data. The results between the other three batch sizes do not differ much, but a choice has been made to use a batch size of 16, as this scores the best on the IoU. A difference in training time was also observed between the four options. Training the model with batch size one took approximately 13 hours, while training the other three models took six to seven hours. However, it should be noted that the models were not run on a GPU, meaning that the training times could potentially be decreased by using different hardware.

Table 14: Comparison of models trained with the augmented data set III with different batch sizes.

Batch size	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
1	0.95	0.60	0.80	0.85	0.28	0.60
8	0.94	0.87	0.88	0.83	0.65	0.79
16	0.93	0.85	0.90	0.89	0.66	0.79
32	0.93	0.87	0.90	0.85	0.65	0.78



(a) Accuracy over 500 epochs.



(b) Loss over 500 epochs.

Figure 47: Plots of the u-net model trained with augmented data set III with a batch size of one, which shows a slower convergence due to a too low learning rate.

7 Conclusion

The aim of this research was to automate the analysis of microscopy images of CFRP laminate cross-sections using deep learning. In this way, manual tuning for each new micrograph, which is done with thresholding techniques, is no longer required. The main goal for analysing these micrographs is to determine the fibre and void volume fractions, as well as the distributions of the fibres. It has been shown that the u-net model is a powerful deep learning architecture for this problem, as it does not require a large number of training images and can learn quickly, even when the hardware used is not optimised for deep learning techniques.

The u-net models trained with the three different data sets show that the performance of a model is highly dependent on the diversity of the training data set. For the model to be able to perform well on all different types of microscopy images with different fibre sizes, lighting settings, and void contents, a high variety of training images is needed. However, there were not many different micrographs available and collecting images from additional micrographs is a time-consuming process. Fortunately, data augmentation has become a viable solution to enhance the variety of microscopy images and increase the robustness of the models.

While the ground truth masks generated may not be completely accurate, the mask generation method is still preferable to manually annotating the images, which is time-consuming or requires expensive software. The models are often able to make predictions that are more accurate than the actual masks of the input image. However, this illustrates the main pitfall of this research; the predictions are evaluated based on comparisons with ground truth masks that already contain errors. Therefore, it is advised not only to focus on accuracy and IoU results, but also to visually inspect the predictions. This visual inspection revealed that the final model usually performs better than the ground truth masks generated.

The fibre and void volume fractions in the ground truth masks are relatively low, as a result of the mask generation process, which may not reflect reality. However, the trained model demonstrates its ability to accurately predict the fibre and void volume fractions, which are in close agreement with the values of the ground truth masks.

Overall, the final u-net model trained with data set III successfully demonstrates that the micrographs used can be well automatically segmented to calculate their fibre and void content with an accuracy of 0.90 and an IoU of 0.79 on the test data set. Furthermore, the centres can be extracted from the predicted output images to determine the fibre distributions. Thus, this is a promising method to replace thresholding methods that need to be manually calibrated for each micrograph, given that the analysed micrographs are partially similar to the training images, and that the ground truth masks represent the input images well.

8 Recommendations

This chapter will discuss the recommendations for further research based on three main components of this study: the data sets, the ground truth masks and the model.

Data sets

As shown in this research, the performance of the u-net model is highly dependent on the quality of the training data set. It was tried to create a data set that was as diverse as possible, data set III, with still having images left to use for the test set. However, only a limited amount of suitable data was available during the time span of this research. Therefore, it is recommended to collect more data from various new micrographs and include other materials, such as glass fibres, to increase the training data set with distinctive training images. This will make the model better equipped to handle a wider range of micrographs. To further test the prediction capabilities of the model on new data, it is also advised to evaluate the model with a larger test set containing images that come from different micrographs as the images present in the training data set.

As collecting new data is a time-consuming process, the option of using synthetic data as training data is also considered. One way to create synthetic data is with a generative adversarial network (GAN), which is a machine learning framework composed of two networks. A generator network that produces new data with the same statistics as the data used for training. And a discriminator network that needs to tell if the data is synthetic or real. The generator is then trained to fool the discriminator so that the generated data will look authentic to human observers. However, GANs usually require between 50,000 and 100,000 training images to get high-quality results. Therefore, this option was not useful in this research, and data augmentation is preferred. However, there is research being done on GANs that only need a few thousand training images by using adaptive discriminator augmentation [44, 45]. This is an interesting option, which could be explored in further research when more training images are collected.

Ground truth masks

The quality of the ground truth masks of the training data can also be improved; however, it is unlikely that the benefit gained from this will be worth the additional time and effort required. Nevertheless, there are two options that are suited to increase the quality without manually labelling every pixel. First, the current mask generation process can be improved by manually checking the predicted fibre centres in ImageJ and removing and adding the ones that are predicted incorrectly. Rather than relying on centre point predictions, the thresholding values can be used to determine which pixels belong to which class. Second, paid software, including AI-assisted labelling, can be used to label images. When money is not an issue, this is the preferred option, because these software programmes are developed to solve annotation problems fast and efficiently.

Model

The current model's accuracy can be enhanced by improving the data and ground truth masks, as well as by altering the model itself. Much research has already been done on the

number of layers and hyperparameters in the model, but the influence of different loss functions and optimisers can be further investigated. The IoU metric has been shown to provide a more accurate assessment of the performance of the model for this task. Therefore, it is interesting to investigate whether a loss function based on this IoU metric can improve the model's performance.

Another recommendation is to improve the semantic segmentation accuracy results using transfer learning. With transfer learning, the knowledge learnt from one task is transferred to a new target task. This is done by using a pre-trained model instead of building one from scratch. Pre-trained models contain pre-trained weights that are used as a starting point. This is possible since lower-level features, such as curves, edges, and circles, are often similar for most computer vision tasks [13]. These primitive features contained in the early layers of a neural network can be easily reused in other arbitrary applications. Pre-trained models are trained with large existing data sets. These models can then be retrained with a significantly smaller data set that contains images for the target task. Therefore, transfer learning from a pre-trained model trained with a large image data set can be beneficial in preventing overfitting, improving accuracy results, and reducing training time on a limited image data set [12, 21]. This makes using a pre-trained model another suitable approach for this semantic segmentation problem with limited data.

9 References

- [1] H. Teng, S. Li, Z. Cao, S. Li, C. Li, and T. J. Ko, “Carbon Fiber Composites for Large-Scale Wind Turbine Blades: Applicability Study and Comprehensive Evaluation in China,” *Journal of Marine Science and Engineering*, vol. 11, no. 3, 2023.
- [2] S. S. P. Reddy, R. Suresh, H. M.B., and B. Shivakumar, “Use of composite materials and hybrid composites in wind turbine blades,” *Materials Today: Proceedings*, vol. 46, pp. 2827–2830, 2021. [Online]. Available: <https://doi.org/10.1016/j.matpr.2021.02.745><https://linkinghub.elsevier.com/retrieve/pii/S2214785321018988>
- [3] E. Cilley, D. Roylance, and N. Schneider, “Methods of Fiber and Void Measurement in Graphite/Epoxy Composites,” in *Composite Materials: Testing and Design (Third Conference)*. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959: ASTM International, 2009, pp. 237–237. [Online]. Available: <http://www.astm.org/doiLink.cgi?STP35492S>
- [4] Bob Lacovara, “Why out of autoclave processing is good for the composites industry,” 2013. [Online]. Available: <https://www.compositesworld.com/columns/why-out-of-autoclave-processing-is-good-for-the-composites-industry>
- [5] S. Paciornik and J. R. D’almeida, “Measurement of void content and distribution in composite materials through digital microscopy,” *Journal of Composite Materials*, vol. 43, no. 2, pp. 101–112, 2009.
- [6] S. Hassani, M. Mousavi, and A. H. Gandomi, “Structural health monitoring in composite structures: A comprehensive review,” *Sensors*, vol. 22, no. 1, pp. 1–45, 2022.
- [7] W. Groupe, “Lecture notes in composites (2. composites and their constituents),” September 2019, university of Twente - Faculty of Engineering Technology.
- [8] M. I. Ali and J. Anjaneyulu, “Effect of fiber-matrix volume fraction and fiber orientation on the design of composite suspension system,” *IOP Conference Series: Materials Science and Engineering*, vol. 455, no. 1, p. 012104, 12 2018. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1757-899X/455/1/012104>
- [9] ASTM, *D3171-09 - Standard Test Methods for Constituent Content of Composite Materials*. West Conshohocken, PA: ASTM International, 2009, no. May 2009. [Online]. Available: <https://www.astm.org/d3171-22.html>
- [10] J. E. Little, X. Yuan, and M. I. Jones, “Characterisation of voids in fibre reinforced composite materials,” *NDT & E International*, vol. 46, no. 1, pp. 122–127, 3 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.ndteint.2011.11.011><https://linkinghub.elsevier.com/retrieve/pii/S0963869511001769>
- [11] J. M. Machado, J. M. R. Tavares, P. P. Camanho, and N. Correia, “Automatic void content assessment of composite laminates using a machine-learning approach,” *Composite Structures*, vol. 288, no. April 2021, 2022.

- [12] L. Luo, B. Zhang, Y. Lei, G. Zhang, Z. Zhang, B. Meng, and Z. Liu, "Identification of voids and interlaminar shear strengths of polymer-matrix composites by optical microscopy experiment and deep learning methodology," *Polymers for Advanced Technologies*, vol. 32, no. 4, pp. 1853–1865, 4 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/pat.5226>
- [13] N. Ketkar and J. Moolayil, *Deep Learning with Python*. Berkeley, CA: Apress, 2021. [Online]. Available: <http://link.springer.com/10.1007/978-1-4842-5364-9>
- [14] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Velasco-Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep Learning vs. Traditional Computer Vision," *Advances in Intelligent Systems and Computing*, vol. 943, no. Cv, pp. 128–144, 10 2019. [Online]. Available: <http://arxiv.org/abs/1910.13796><http://dx.doi.org/10.1007/978-3-030-17795-9>
- [15] S. Liu, *Zhi-Hua Zhou*, 2021. [Online]. Available: <https://link.springer.com/10.1007/978-981-15-1967-3>
- [16] E. F. Morales and J. H. Zaragoza, "An introduction to reinforcement learning," *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, pp. 63–80, 2011.
- [17] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, no. December 2020, p. 100057, 7 2021. [Online]. Available: <https://doi.org/10.1016/j.array.2021.100057><https://linkinghub.elsevier.com/retrieve/pii/S2590005621000059>
- [18] Z. B. Kizilkan, M. S. Sivri, I. Yazici, and O. F. Beyca, "Neural Networks and Deep Learning," in *Springer Series in Advanced Manufacturing*, 2022, pp. 127–151. [Online]. Available: https://link.springer.com/10.1007/978-3-030-93823-9_5
- [19] S. A. Cambo and D. Gergle, "User-Centred Evaluation for Machine Learning," 2018, pp. 315–339. [Online]. Available: http://link.springer.com/10.1007/978-3-319-90403-0_16
- [20] The MathWorks, "Introducing Deep Learning with MATLAB," 2021. [Online]. Available: https://www.mathworks.com/content/dam/mathworks/ebook/gated/80879v00_Deep_Learning_ebook.pdf?s_tid=wtest_ctent_HTML
- [21] C. Chen, C. Qin, H. Qiu, G. Tarroni, J. Duan, W. Bai, and D. Rueckert, "Deep learning for cardiac image segmentation: A review," *Frontiers in Cardiovascular Medicine*, vol. 7, no. March, 11 2019. [Online]. Available: <http://arxiv.org/abs/1911.03723><http://dx.doi.org/10.3389/fcvm.2020.00025>
- [22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015. [Online]. Available: <https://www.nature.com/articles/nature14539>
- [23] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 10 2016. [Online]. Available: <http://arxiv.org/abs/1610.02357>

- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [25] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *32nd International Conference on Machine Learning, ICML 2015*, vol. 1, pp. 448–456, 2 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [26] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *IEEE Access*, 2015, vol. 9, pp. 234–241. [Online]. Available: https://ieeexplore.ieee.org/document/9330594/http://link.springer.com/10.1007/978-3-319-24574-4_28
- [27] J. Zhang, T. Xie, C. Yang, H. Song, Z. Jiang, G. Zhou, D. Zhang, H. Feng, and J. Xie, “Segmenting purple rapeseed leaves in the field from UAV RGB imagery using deep learning as an auxiliary means for nitrogen stress detection,” *Remote Sensing*, vol. 12, no. 9, 2020.
- [28] T. Rouwmaat, “Microscopy data from C/LM-PAEK as-recieved tapes,” ThermoPlastic composites Research Center.
- [29] E. R. Pierik, “Microscopy data from C/PEEK and C/LM-PAEK friction specimens,” ThermoPlastic composites Research Center.
- [30] L. F. G. Camacho, “Faculty of Engineering Technology Towards a robust in-situ AFP process : Quantifying the effect of surface consolidation state and its correlation with laser power , process temperature , and interlaminar bond strength,” 2023.
- [31] M. Fernhout, “Annotated images of microscopy images from carbon fibre composites,” <https://data.4tu.nl/datasets/bf11171a-1f57-4a8a-9768-757850d13665>, October 2023, doi: 10.4121/bf11171a-1f57-4a8a-9768-757850d13665.v1.
- [32] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, “NIH Image to ImageJ: 25 years of image analysis,” *Nature Methods*, vol. 9, no. 7, pp. 671–675, 7 2012. [Online]. Available: <https://www.nature.com/articles/nmeth.2089>
- [33] I. V. Grishagin, “Automatic cell counting with ImageJ,” *Analytical Biochemistry*, vol. 473, pp. 63–65, 3 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.ab.2014.12.007https://linkinghub.elsevier.com/retrieve/pii/S0003269714005570>
- [34] T. Ferreira and W. Rasband, “ImageJ User Guide,” 2010. [Online]. Available: <https://imagej.net/ij/docs/guide/index.html>
- [35] C. Glasbey, “An Analysis of Histogram-Based Thresholding Algorithms,” *Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532–537, 11 1993. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/doi/10.1006/gmip.1993.1040>

- [36] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim, K. M. Hasib, S. K. Patel, M. Jonkman, and Z. I. Mannan, “MNet-10: A robust shallow convolutional neural network model performing ablation study on medical images assessing the effectiveness of applying optimal data augmentation technique,” *Frontiers in Medicine*, vol. 9, no. August, 8 2022. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fmed.2022.924979/full>
- [37] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, “Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks,” pp. 1–12, 5 2017. [Online]. Available: <http://arxiv.org/abs/1705.03820>
- [38] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and Flexible Image Augmentations,” *Information*, vol. 11, no. 2, p. 125, 2 2020. [Online]. Available: <https://www.mdpi.com/2078-2489/11/2/125>
- [39] A. M. Reza, “Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement,” *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 38, no. 1, pp. 35–44, 8 2004. [Online]. Available: <http://link.springer.com/10.1023/B:VLSI.0000028532.53893.82>
- [40] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 12 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [41] M. Berman, A. R. Triki, and M. B. Blaschko, “The Lov\`asz-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4413–4421, 5 2017. [Online]. Available: <http://arxiv.org/abs/1705.08790>
- [42] Y. Wu, L. Liu, J. Bae, K.-H. Chow, A. Iyengar, C. Pu, W. Wei, L. Yu, and Q. Zhang, “Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks,” in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 12 2019, pp. 1971–1980. [Online]. Available: <http://arxiv.org/abs/1908.06477><https://ieeexplore.ieee.org/document/9006104/>
- [43] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700 LECTU, pp. 437–478, 6 2012. [Online]. Available: <http://arxiv.org/abs/1206.5533>
- [44] I. Salián, “Nvidia research achieves ai training breakthrough using limited datasets,” <https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/#:~:text=It%20typically%20takes%2050%2C000%20to,falter%20at%20producing%20realistic%20results.>, accessed: 24-09-2023.

- [45] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” *Advances in Neural Information Processing Systems*, vol. 2020-Decem, no. NeurIPS, 6 2020. [Online]. Available: <http://arxiv.org/abs/2006.06676>
- [46] E. Pierik, W. Groupe, S. Wijskamp, and R. Akkerman, “Modeling the effect of temperature and pressure on the peak and steady-state ply–ply friction response for UD C/PAEK tapes,” *Composites Part A: Applied Science and Manufacturing*, vol. 173, no. June, p. 107671, 10 2023. [Online]. Available: <https://doi.org/10.1016/j.compositesa.2023.107671><https://linkinghub.elsevier.com/retrieve/pii/S1359835X23002476>

Appendices

A Micrograph details

In this appendix, the origin of the different micrographs and their sources are explained in detail. In total, 30 micrographs from five different sources were used to create the 14 image sets. Table 15 shows an overview of all micrographs and the image set to which they belong. Also, from which specimen the micrographs are taken and the number of 256 x 256 images that have been acquired. Note that each 256 x 256 images in the online data set [31] can be recovered by means of the name of the micrograph as shown in this table. As the name of each image starts with the name of the micrograph image in the form NAME_Y_X. Where X and Y are locations of where the 256 x 256 image is taken from in the larger original micrograph.

Table 15: All 30 micrographs used in this research and their corresponding images sets and number of images.

Micrograph	Image set	Image source	Zoom	Specimen	Number of training images	Number of images with voids
2_3_1_R	PEEK 3A	1	200x	1-A	72	64
2_3_2_L	PEEK 3A	1	200x	1-A	70	58
2_3_2_R	PEEK 3A	1	200x	1-A	72	65
2_3_3_R	PEEK 3A	1	200x	1-A	66	52
2_6_3_R	PEEK 3A	1	200x	1-B	76	58
2_6_1_R	PEEK 3B	1	200x	1-B	68	422
2_6_2_R	PEEK 3C	1	200x	1-B	76	58
2-6-1_mid	PEEK 7	1	400x	1-B	16	4
17-5-2_zoom_mid	LM-PAEK 7	1	400x	1-C	16	2
0_press_375_R2_300x_xsf	LM-PAEK 5	2	300x	2-press-A	100	0
0_ISC_400_R2	LM-PAEK 10	2	500x	2-ISC	137	0
0_press_375_R2_500x_xs_L	LM-PAEK 10	2	500x	2-press-A	36	0
0_press_375_R2_500x_xs_R	LM-PAEK 10	2	500x	2-press-A	30	0
90_press_375_R4_500x_surfxs_R	LM-PAEK 10	2	500x	2-press-B	47	0
specimen_33_left	LM-PAEK 14	3	500x	3-A	8	0
specimen_33_middle	LM-PAEK 14	3	500x	3-A	12	0
specimen_33_right	LM-PAEK 14	3	500x	3-A	7	0
specimen_35_left	LM-PAEK 14	3	500x	3-B	10	0
specimen_35_middle	LM-PAEK 14	3	500x	3-B	16	0
specimen_35_right	LM-PAEK 14	3	500x	3-B	8	0
specimen_37_left	LM-PAEK 14	3	500x	3-C	13	0
specimen_37_middle	LM-PAEK 14	3	500x	3-C	16	0
specimen_37_right	LM-PAEK 14	3	500x	3-C	9	0
specimen_43_middle	LM-PAEK 14	3	500x	3-D	1	0
G02_1	PEEK 2	4	100x	4-A	100	42
G02_2	PEEK 3D	4	200x	4-A	100	19
G02_3	PEEK 4	4	500x	4-A	100	36
Start_M	LM-PAEK 6	5	300x	5-A	100	9
Stop_R	LM-PAEK 11	5	500x	5-A	100	7
Middle_R	LM-PAEK 22	5	800x	5-A	100	4

A.1 Source 1

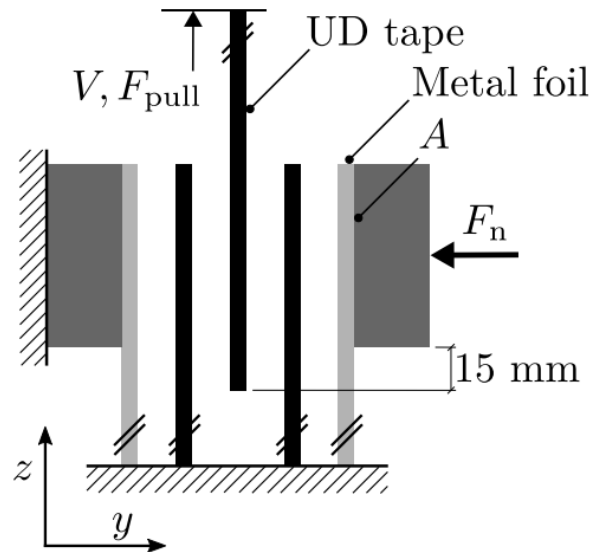


Figure 48: Schematic overview of the friction tester used [46].

From this source, nine different micrographs are used to create five different image sets. These micrographs are taken from three different specimens: 1-A, 1-B, and 1-C, as shown in Table 15. Specimens 1-A and 1-B are made from three UD plies of semi-crystalline PEEK, Toray TC1200. And specimen 1-C is made of Toray TC1225 UD carbon LM-PAEK. All specimens are tested on a friction tester before being analysed with a microscope. The setup used in the friction tester is shown in Figure 48 and is used in the research by E.R. Pierik [46] to characterise the ply-ply friction response of the UD tape. This is done by forcing the central ply, which is clamped at the upper clamp of the testing machine, to slide against the two stationary outer plies that are clamped at the bottom, which is done at a constant rate. Specimen 1-A is tested at a speed of 25 mm/min on the friction tester, and specimens 1-B and 1-C are tested at a speed of 5 mm/min. All of the carbon fibres used are aligned in the longitudinal direction. This is also the sliding direction in which the specimens are tested in the friction tester. After these tests, the two specimens are cut at three locations where the three plies meet, for example, the location indicated in Figure 49, to be able to analyse different cross-sections with a Keyence digital microscope.

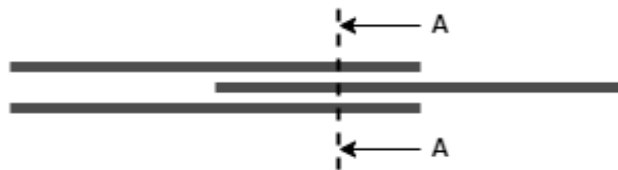


Figure 49: Schematic overview of a location where a cut is made in the 3-ply laminate to create a micrograph of the cross-sectional area [46].

A.2 Source 2

Source 2 provided five micrographs that were used to create two image sets, as shown in Table 15. These micrographs came from research toward a robust automated in situ fibre placement process by Luis F. Conzalez Camacho [30]. The four different micrographs are made from three distinct specimens. All specimens are made from unidirectional carbon fibre reinforced polymers: TC1225 low-melting polyaryletherketone (LM-PAEK) from Toray. The specimens consist of 12 unidirectional layers with an alternating stacking of 0° and 90° layers and finally an additional unidirectional tape on top of the specimens which was of interest for the specific research. Specimens 2-press-A and 2-press-B are fully consolidated through a press cycle. Whilst sample 250-ISC is partially consolidated through a press cycle, but with the extra top layer in-situ consolidated.

A.3 Source 3

Source 3 comprised 10 micrographs of specimen 3-A, 3-B, 3-C and 3-D, shown in Table 15, which were analysed with a microscope to investigate the adhesion between thermoplastic tapes by Thijs Rouwmaat [28]. The cross-sections are made from the one-layer “as received” tapes from Toray TC1225 UD C/LM-PAEK.

A.4 Source 4

Source 4 contains three micrographs made from specimen 4-A. Micrographs are created specifically for this research and are made with different zoom settings to create more diverse data. Thus, three different image sets were created, as shown in Table 15. The consolidated laminate used was made from carbon-reinforced PEEK(Toray TC1200) with a $[0]_{16s}$ lay-up.

A.5 Source 5

With source 5, again three image sets are created with three micrographs. The micrographs were made specially for this research with different zoom settings, as can be seen in Table 15, and were obtained from specimen 5-A. This specimen is a 3-ply UD laminate with a $[0/0/0]$ lay-up, which was made with fibre placement and in-situ consolidation. The specimen was cut at three different locations, indicated by “Start”, “Middle”, and “Stop” in the name of the micrographs. With the assumption that more voids would appear at the beginning and end of the laminate when the fibre placement method did not work optimally. However, this was not observed when analysing the micrographs. Unfortunately, there were no significant differences in the number of voids at the three locations and there were fewer voids than expected.

B Data sets

Here, a more detailed structure of each of the data sets presented in Chapter 3.4 is given. Tables 16 to 19 indicate which micrographs are used for each data set and how many images are taken of each source. The data sets are available online at the 4TU.ResearchData data repository [31].

Table 16: Composition of data set I from seven stitched micrographs.

Micrograph	Image set	Image source	Zoom	Specimen	Number of training images	Number of images with voids
2_3_1_R	PEEK 3A	1	200x	1-A	72	64
2_3_2_L	PEEK 3A	1	200x	1-A	70	58
2_3_2_R	PEEK 3A	1	200x	1-A	72	65
2_3_3_R	PEEK 3A	1	200x	1-A	66	52
2_6_1_R	PEEK 3B	1	200x	1-B	68	42
2_6_2_R	PEEK 3C	1	200x	1-B	76	58
2_6_3_R	PEEK 3A	1	200x	1-B	76	66

Table 17: Composition of data set II from eight different micrographs.

Micrograph	Image set	Image source	Zoom	Specimen	Number of training images	Number of images with voids
2_3_1_R	PEEK 3A	1	200x	1-A	72	64
2_3_2_R	PEEK 3A	1	200x	1-A	36	30
2_3_3_R	PEEK 3A	1	200x	1-A	66	52
2_6_2_R	PEEK 3C	1	200x	1-B	76	58
0_ISC_400_R2	LM-PAEK 10	2	500x	2-ISC	137	0
0_press_375_R2_500x_xs_L	LM-PAEK 10	2	500x	2-press-A	36	0
0_press_375_R2_500x_xs_R	LM-PAEK 10	2	500x	2-press-A	30	0
90_Press_375_R4_500x_surfxs_R	LM-PAEK 10	2	500x	2-press-B	47	0

Table 18: Composition of data set III from 22 different micrographs.

Micrograph	Image set	Image source	Zoom	Specimen	Number of training images	Number of images with voids
2_3_1_R	PEEK 3A	1	200x	1-A	25	20
2_3_2_L	PEEK 3A	1	200x	1-A	25	20
2_3_3_R	PEEK 3A	1	200x	1-A	25	20
2_6_1_R	PEEK 3B	1	200x	1-B	25	14
0_ISC_400_R2	LM-PAEK 10	2	500x	2-ISC	100	0
0_press_375_R2_300x_xsf	LM-PAEK 5	2	300x	2-press-A	100	0
specimen_33_left	LM-PAEK 14	3	500x	3-A	8	0
specimen_33_middle	LM-PAEK 14	3	500x	3-A	12	0
specimen_33_right	LM-PAEK 14	3	500x	3-A	7	0
specimen_35_left	LM-PAEK 14	3	500x	3-B	10	0
specimen_35_middle	LM-PAEK 14	3	500x	3-B	16	0
specimen_35_right	LM-PAEK 14	3	500x	3-B	8	0
specimen_37_left	LM-PAEK 14	3	500x	3-C	13	0
specimen_37_middle	LM-PAEK 14	3	500x	3-C	16	0
specimen_37_right	LM-PAEK 14	3	500x	3-C	9	0
specimen_43_middle	LM-PAEK 14	3	500x	3-D	1	0
G02_1	PEEK 2	4	100x	4-A	100	42
G02_2	PEEK 3D	4	200x	4-A	100	19
G02_3	PEEK 4	4	500x	4-A	100	36
Start_M	LM-PAEK 6	5	300x	5-A	100	9
Stop_R	LM-PAEK 11	5	500x	5-A	100	7
Middle_R	LM-PAEK 22	5	800x	5-A	100	4

Table 19: Composition of the test data set from two different micrographs.

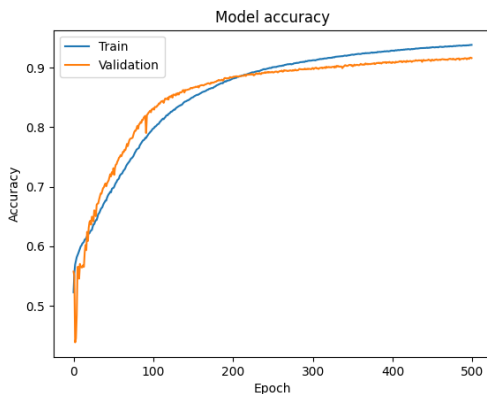
Micrograph	Image set	Image source	Zoom	Specimen	Number of training images	Number of images with voids
2-6-1_mid	PEEK 7	1	400x	1-B	16	4
17-5-2_zoom_mid	LM-PAEK 7	1	400x	1-C	16	2

C Results of models trained with non-augmented data sets

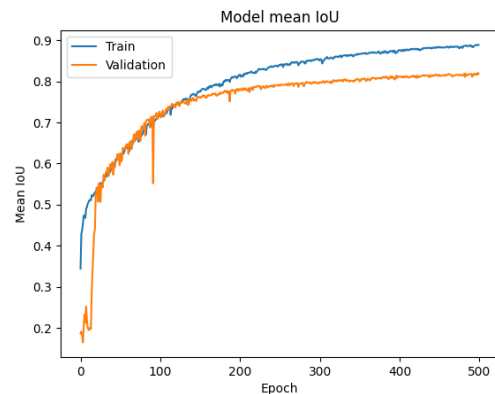
C.1 Data set I

Data set I with non-augmented images is used to train the network, as it contains a lot of images with voids. The model is trained for 500 epochs. As can be seen in Figure 50 the accuracy, mean IoU and loss are again plotted during training for the train and validation set.

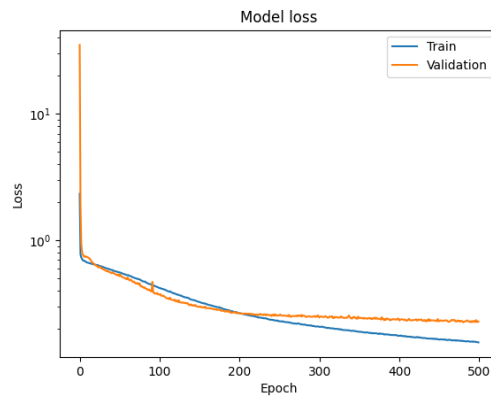
Figure 50a shows that a train accuracy of almost 0.94 is reached and a validation accuracy of around 0.92. An example of how a prediction looks on a validation image is shown in Figure 51. In Figure 51d yellow pixels indicate pixels that are predicted as voids, but are matrix material in the ground truth mask. Pink pixels are the opposite; pixels that are predicted as matrix material, but belong to voids in the ground truth masks. However, in this example, only two pink pixels are present, and are thus barely visible.



(a) Categorical accuracy over 500 epochs.



(b) Mean IoU over 500 epochs.



(c) Loss over 500 epochs.

Figure 50: Evaluation plots of the trained u-net model with data set I including voids.

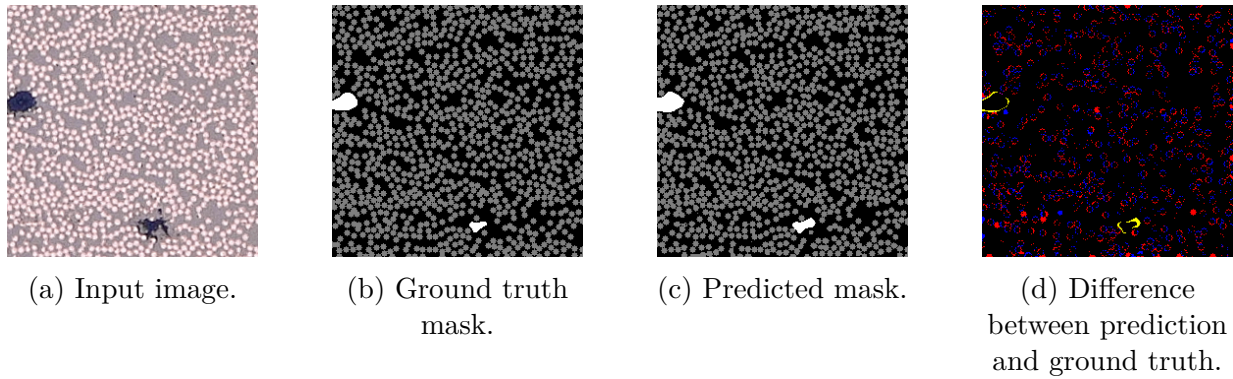


Figure 51: Prediction of the trained u-net model with data set I with voids for an example image with an accuracy of 0.92 and an IoU of 0.82.

The model that achieved the best results during training is used to make predictions on the test data. The average accuracy achieved is 0.67 and the mean IoU is 0.32. This mean IoU is lower than the previous model, which is expected because of the higher complexity of the images including voids. An example prediction of an image from the test set is shown in Figure 52. As can be seen, the same problem arises with the fibre sizes of the predicted image as with the model without voids. However, the void is predicted quite well, a bit larger than it actually is. But the model also predicted an extra void in the middle, where fewer fibres are present, which is not actually there. It could be that it learnt to indicate large spots where no fibres are present as voids, which is undesirable. In conclusion, this trained model can easily predict voids in the train and validation data. With test data, voids can also be identified, however, there is also a risk of false positives being detected. Now a more diverse data set needs to be used to overcome the problem of incorrectly predicting fibre sizes.

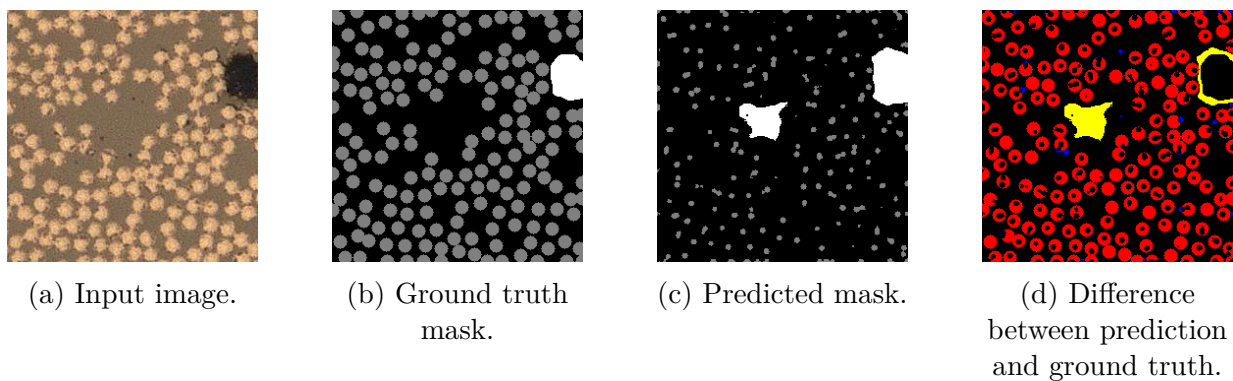
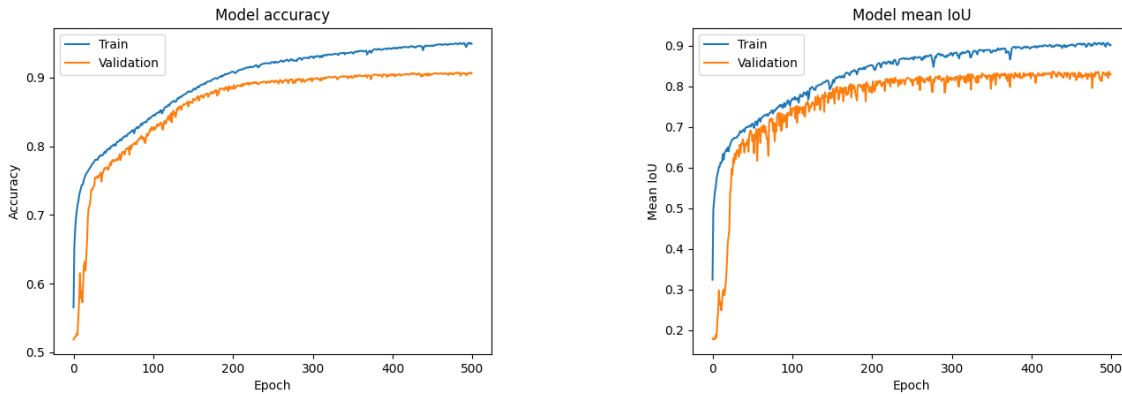


Figure 52: Prediction of the trained u-net model with data set I with voids for a test image with a void with an accuracy of 0.67 and an IoU of 0.42.

C.2 Data set II

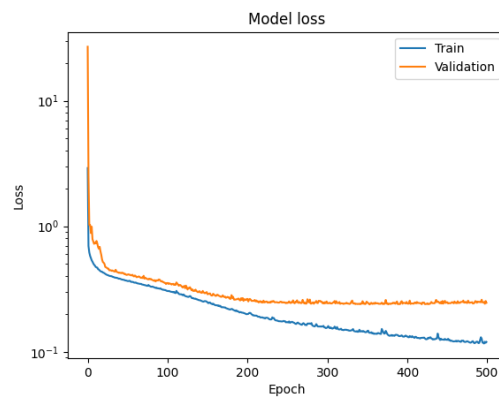
Next, data set II (without data augmentation) is used to train the u-net model. After 500 epochs, the loss has converged to a training value of 0.12 and a validation value of 0.25. The trained model achieved an accuracy of 0.95 on the training set and 0.91 on the validation

set. And an IoU of 0.90 on the training set and 0.83 on the validation set. These results are comparable to the previous two trained models. And, as can be seen in Figures 54 and 55, the images of both data subsets are well predicted.



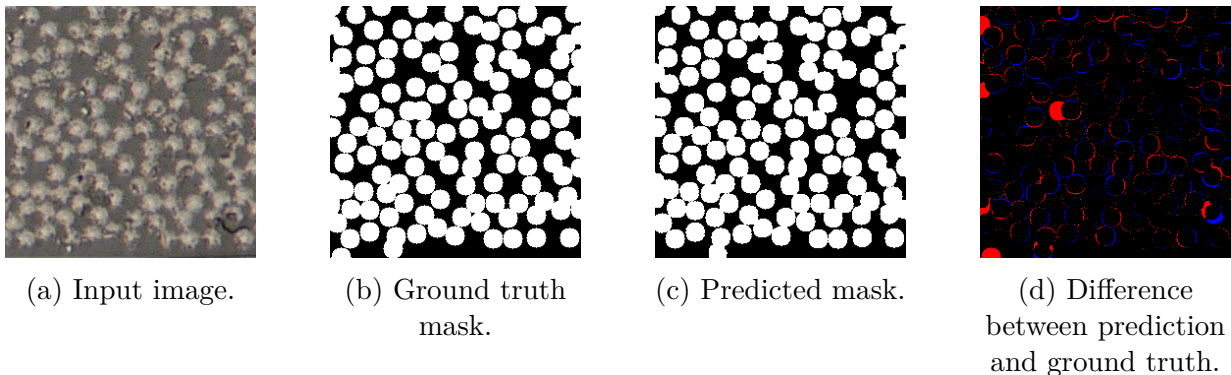
(a) Categorical accuracy over 500 epochs.

(b) Mean IoU over 500 epochs.



(c) Loss over 500 epochs.

Figure 53: Evaluation plots of the trained u-net model with data set II.



(a) Input image.

(b) Ground truth mask.

(c) Predicted mask.

(d) Difference between prediction and ground truth.

Figure 54: Prediction of the trained u-net model with data set II with voids for an example image with an accuracy of 0.95 and an IoU of 0.90.

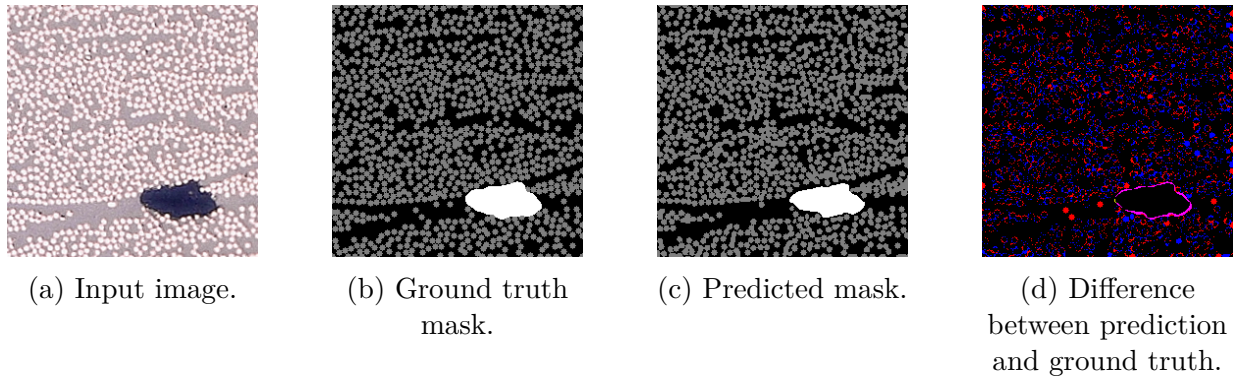


Figure 55: Prediction of the trained u-net model with data set II with voids for an example image with an accuracy of 0.86 and an IoU of 0.78.

However, it is interesting to see if this model performs better on the test data set. The model that performed best during training is used on the test data. This gives a mean accuracy of 0.65 and a mean IoU of 0.48 on the 32 images. The IoU actually improved significantly from the models trained with data set I. However, as can be seen in Figure 56, the fibre size is still not correctly predicted. The fibres are predicted to be too large because of the size of the larger fibres present in the data set. The fibres present in the training data have a radius of three and 10 pixels. However, the fibres in the test data have a fibre size of seven pixels. Thus, it can be concluded that the model is not yet able to interpolate fibre sizes with these training data. Therefore, more variation is needed in the training data, which will be done by using data set III in Appendix C.3.

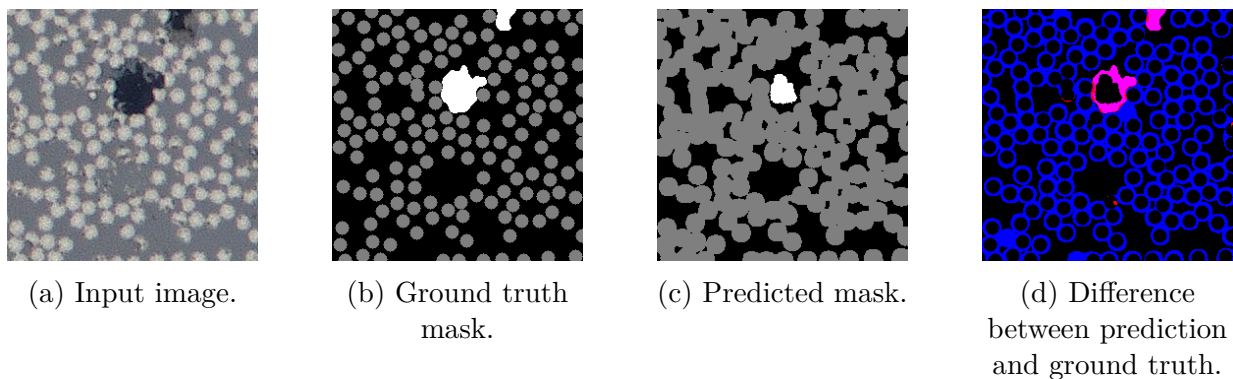


Figure 56: Prediction of the trained u-net model with data set II for an test image with a void with an accuracy of 0.65 and an IoU of 0.44.

C.3 Data set III

This u-net model is trained with the most diverse data set containing 1000 images. As can be seen in Figure 57 the model converges earlier than the previously trained models. After 200 epochs, the validation accuracy, mean IoU and loss are converged to training values of respectively 0.95, 0.87 and 0.12 and validation values of 0.92, 0.78 and 0.27.

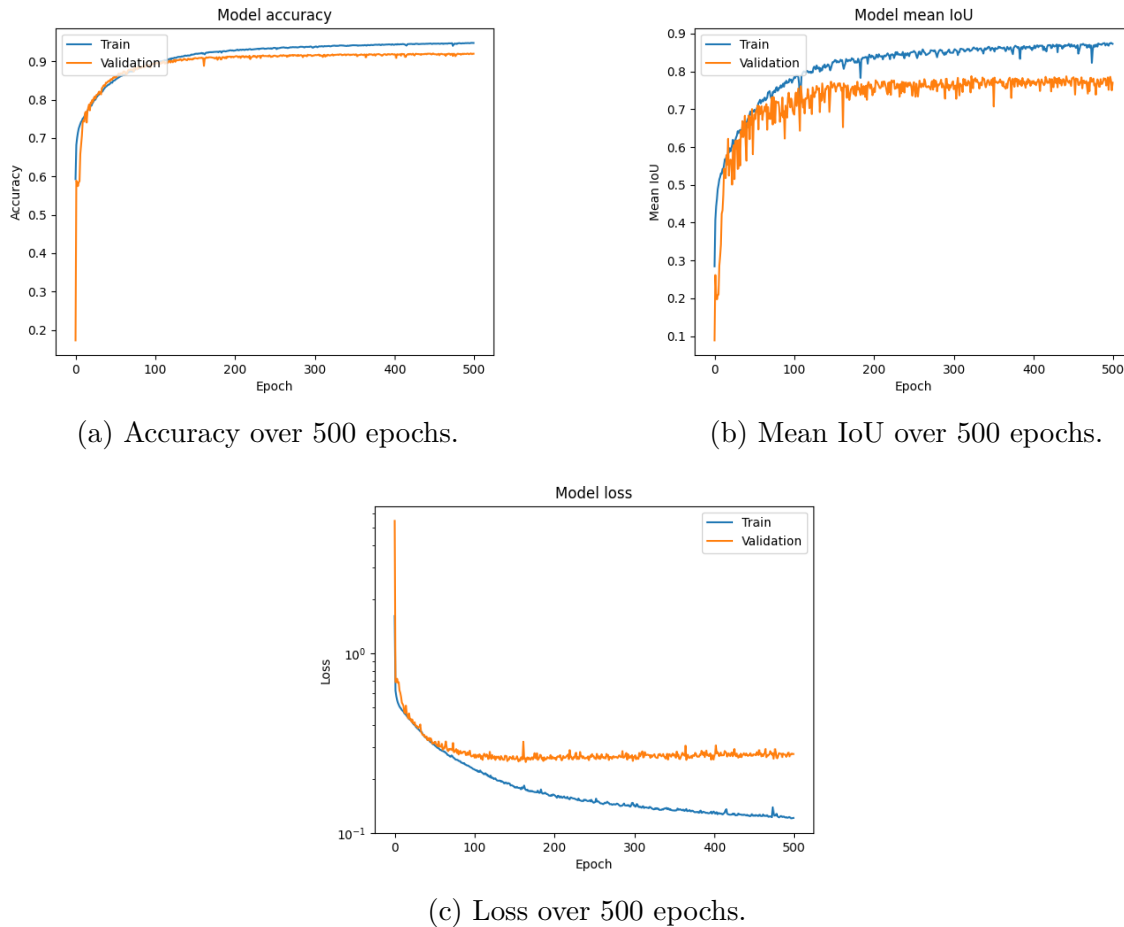


Figure 57: Plots of the u-net model trained with data set III.

The accuracy results are comparable to those of the previous models; however, the IoU is lower than that of these models. This can be explained by the model being worse at predicting voids in images with voids from image sets that contained a relatively low number of voids. An example of such a prediction of an image with a void is shown in Figure 58. Here, the small void in the bottom left corner is not predicted in Figure 58c. Note that in this figure, the white colour indicates the class “fibres” while in Figure 58b the white colour represents the “voids” class and the grey represents the “fibres” class. Because this one class is completely mispredicted, the IoU becomes lower. The model is able to predict bigger voids as can be seen in Figure 59. However, the model does predict the void smaller than in the ground truth mask, resulting in an IoU of 0.72. The best prediction of this validation data is shown in Figure 60 and has an accuracy of 0.96 and an IoU of 0.92.

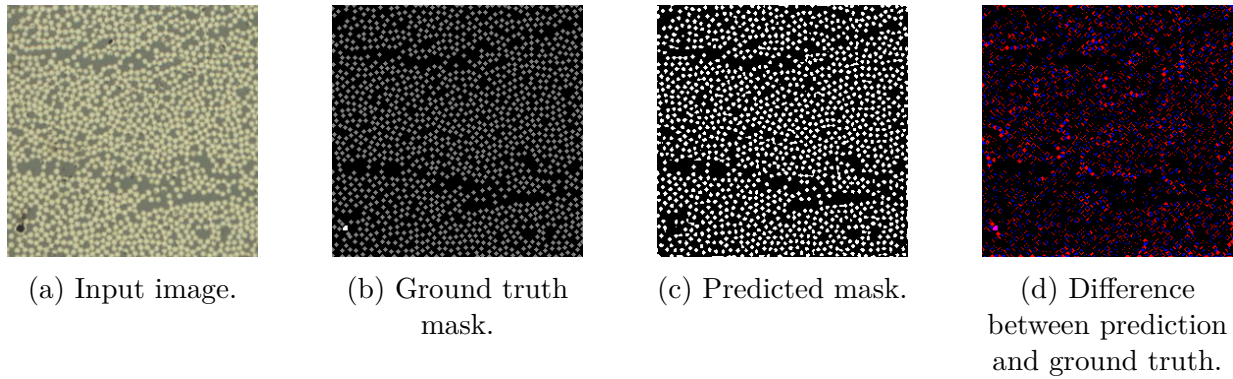


Figure 58: Prediction of the trained u-net model with data set III for an example image with an accuracy of 0.86 and an IoU of 0.47.

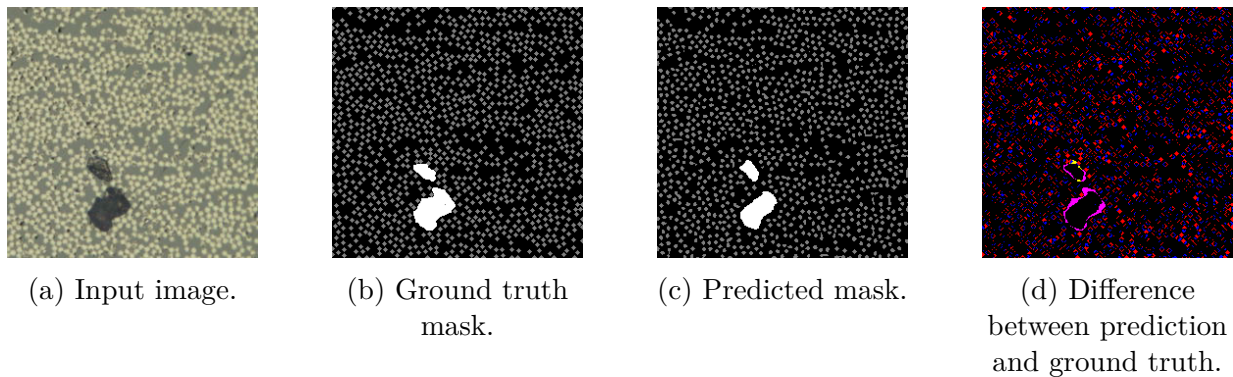


Figure 59: Prediction of the u-net model trained with data set III for an example image with voids and an accuracy of 0.87 and an IoU of 0.72.

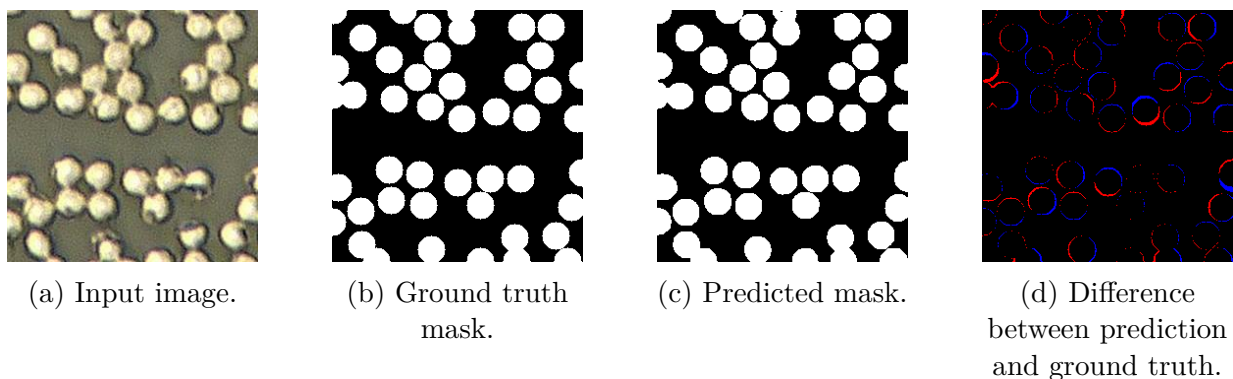


Figure 60: The best prediction from the test set of the u-net model trained with data set III with an accuracy of 0.96 and an IoU of 0.92.

The u-net model that achieved the best performance during training is used on the validation data set of 192 images. The average accuracy achieved is 0.65, and the mean IoU is 0.85. Unfortunately, these results are the same as the model achieved while training with the II

data set. Therefore, increasing the diversity by adding images from more image sets did not improve the test results. This may be because the fibre size of the test data is still not present in the training data and the data is still not diverse enough to handle images with different spectra. In Figure 62 another prediction is shown on the test data that includes a void. This void is predicted to be smaller than it is in the ground truth image, which was also noticed in the validation training set. However, the model is able to correctly recognise the images with voids in the test set. In conclusion, to improve the test results, the data set 1000 is not good enough and needs to be improved by data augmentation.

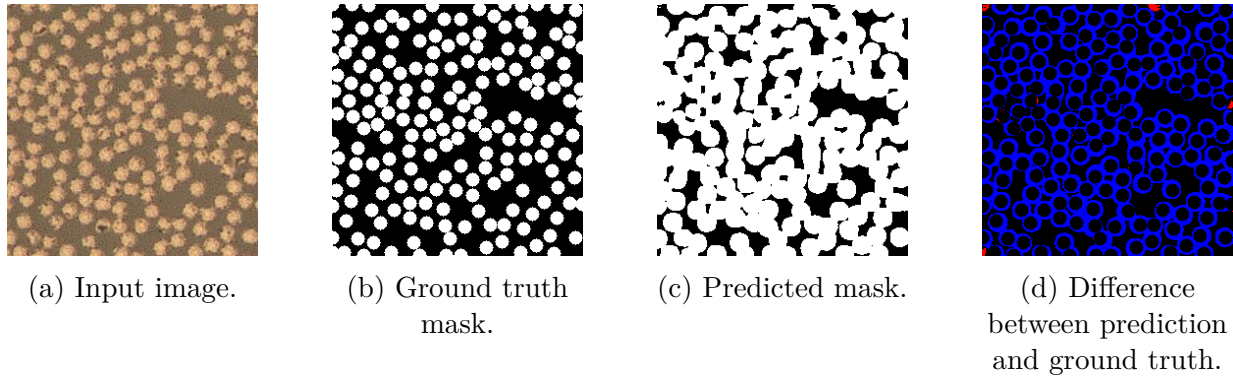


Figure 61: Prediction of the trained u-net model with data set III for an test image with an accuracy of 0.65 and an IoU of 0.48.

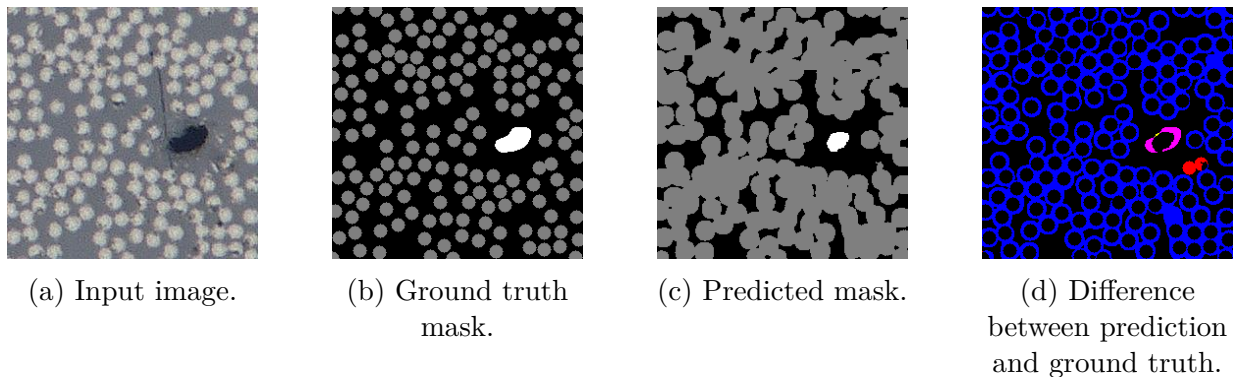


Figure 62: Prediction of the trained u-net model with data set III for an test image with a void with an accuracy of 0.63 and an IoU of 0.47.

D Comparison of model III and the original u-net model

The architecture used in the research is based on the original u-net of Ronneberger et al. [26] and adapted as explained in Chapter 4.1. To illustrate that the adaptations are an improvement of the model for this specific segmentation problem, both models are compared. The accuracy and IoU results are shown in Table 20. Both models achieve approximately the same results with the training and validation data. However, a significant difference is

seen in the test results. The original model did not find a general solution that works as well as the adapted model designed for this research. It is also noteworthy that the original model took 55 hours to train for 500 epochs, while training the final model of this research only took six hours.

Table 20: The accuracy and IoU results of the original model as proposed by Ronneberger et al. [26] trained with data set III and the final adapted model of this research.

Model	Accuracy			IoU		
	Training	Validation	Test	Training	Validation	Test
U-net by Ronneberger et al. [26] Augmented data set III	0.93	0.86	0.68	0.88	0.70	0.34
Augmented data set III	0.93	0.85	0.90	0.89	0.66	0.79

E One fibre prediction

A test is performed to determine if the model trained with data set III is capable of recognising a single fibre when the model is trained with only images containing multiple fibres. From an example image, all fibres are deleted except one, using Photoshop. Figure 63a shows the result that is used as the input image. The predicted mask of the model has an accuracy of 0.997 and an IoU of 0.88 and is shown in Figure 63. This shows that the model can also make a good prediction on one fibre.

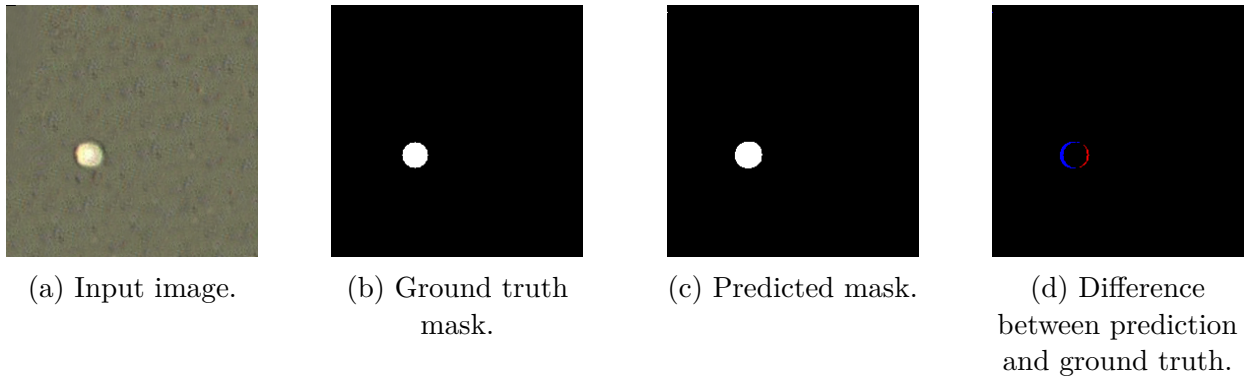


Figure 63: Prediction of the u-net model trained with augmented data set III for an example image with only one fibre with an accuracy of 0.997 and an IoU of 0.88.

F Feature maps final model

This appendix provides an enlarged view of the feature maps from Figure 10 in Chapter 2.7. The input image used for these visualisations is shown in Figure 64. Figure 65 to 73 show examples of feature maps of the specific layers with this input image. Finally, the output, the predicted mask of the model is shown in Figure 74.

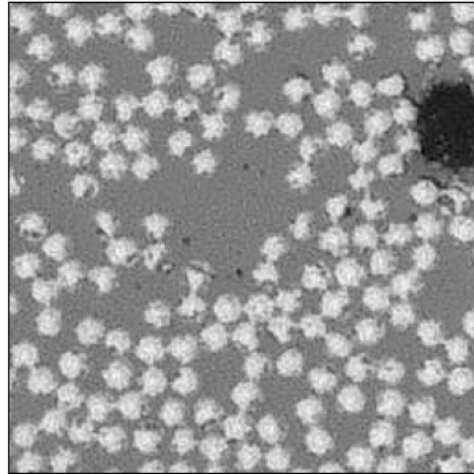


Figure 64: Input image used to visualise the feature maps of the final model trained with augmented data set III.

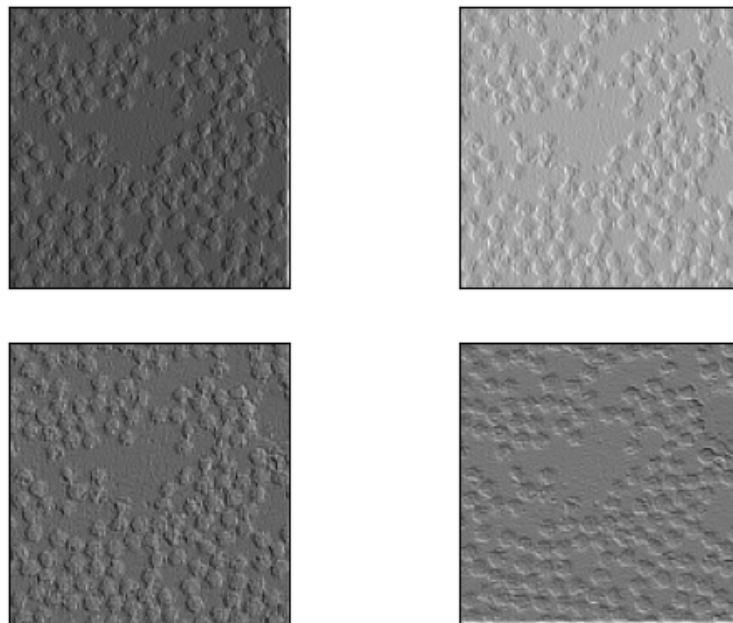


Figure 65: Four random feature maps of the first 3 x 3 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

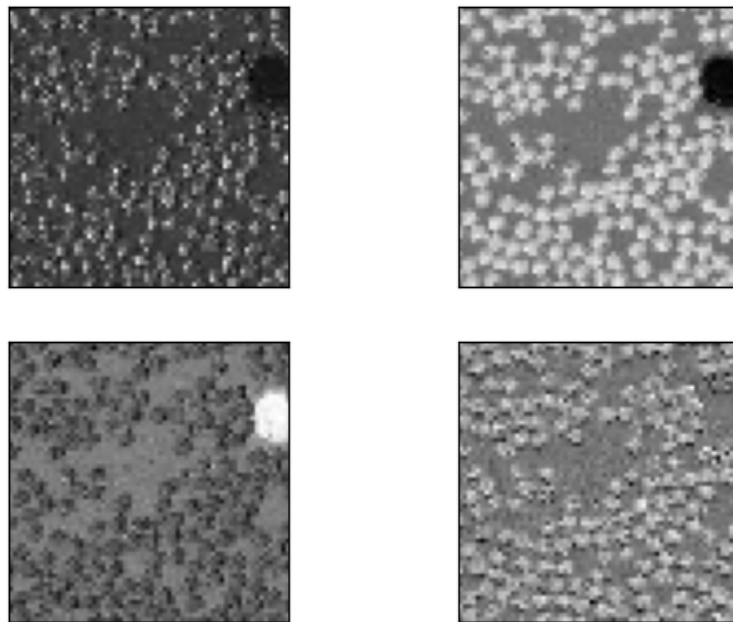


Figure 66: Four random feature maps of the first 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

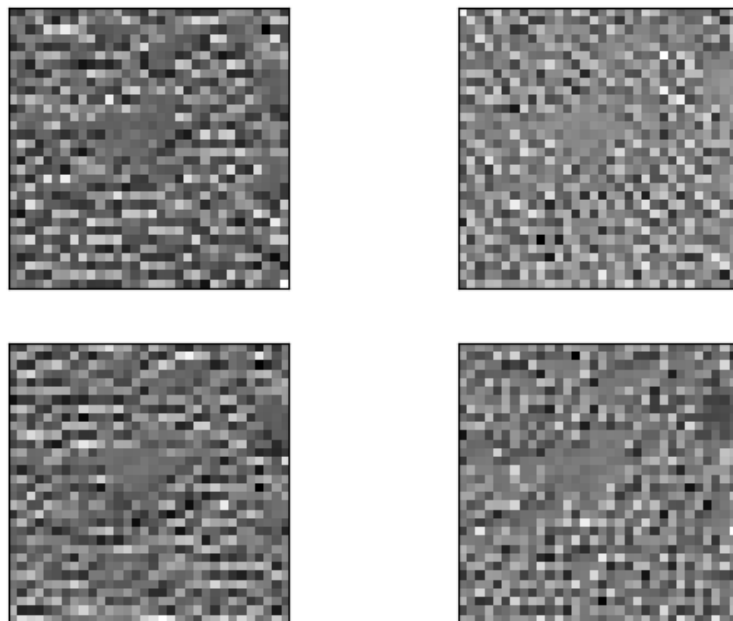


Figure 67: Four random feature maps of the second 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

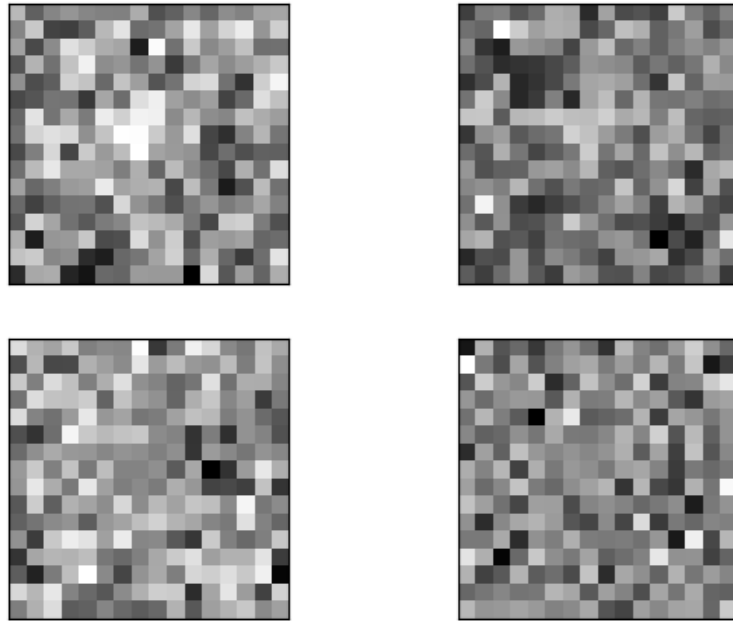


Figure 68: Four random feature maps of the third 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

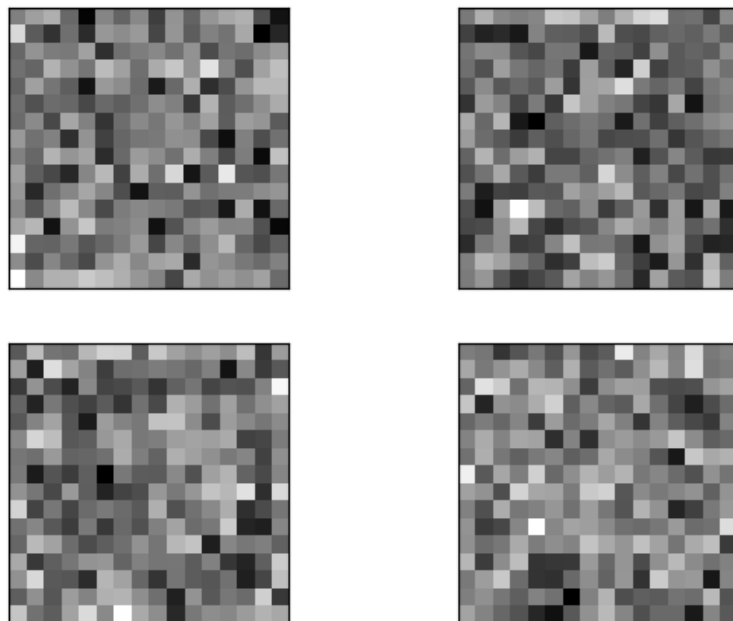


Figure 69: Four random feature maps of the fourth 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

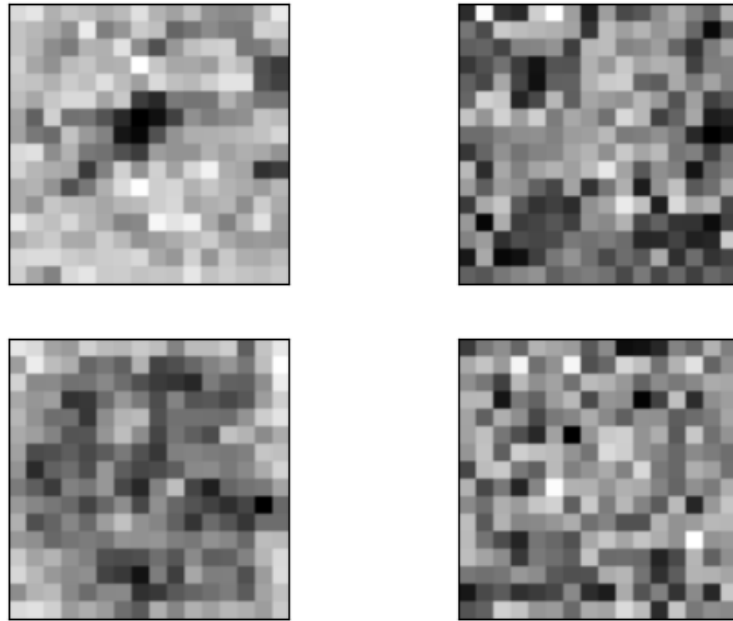


Figure 70: Four random feature maps of the fifth 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

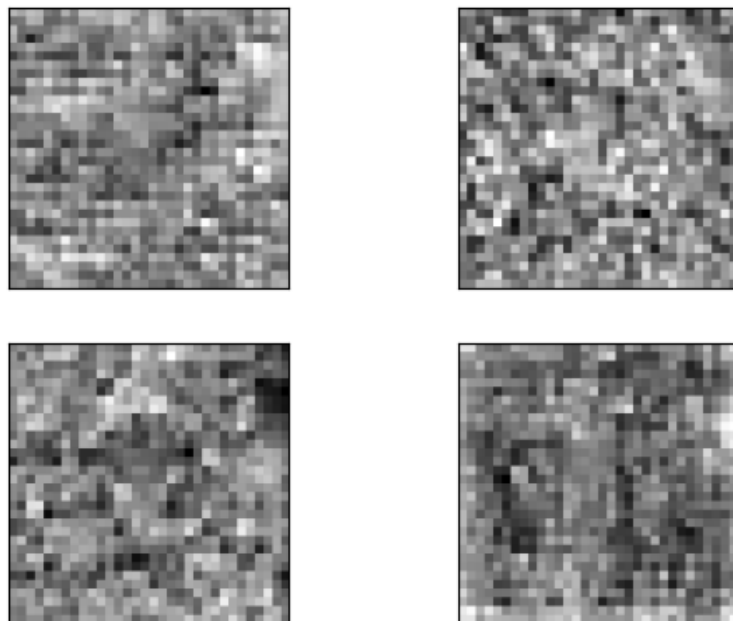


Figure 71: Four random feature maps of the sixth 1×1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.

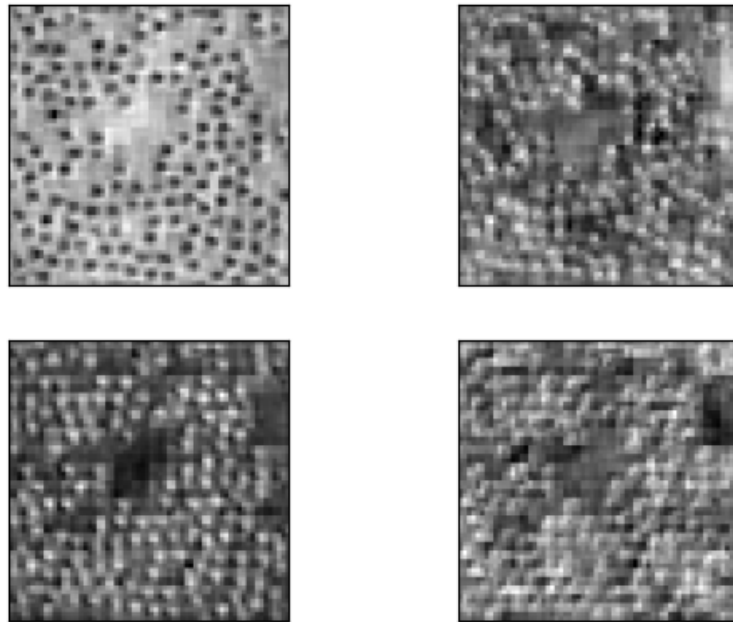
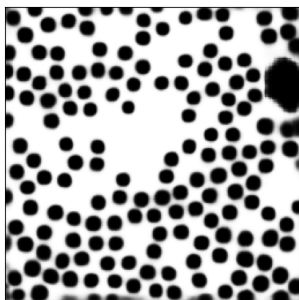
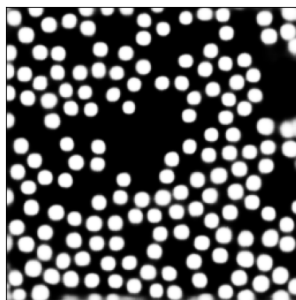


Figure 72: Four random feature maps of the last 1 x 1 convolutional layer of the model trained with augmented data set III visualised with Figure 64 as input image.



(a) Feature map for the first class: matrix material.



(b) Feature map for the second class: fibres.



(c) Feature map for the third class: voids.

Figure 73: The three feature maps of the last 3 x 3 convolutional layer with a softmax activation function of the model trained with augmented data set III visualised with Figure 64 as input image.

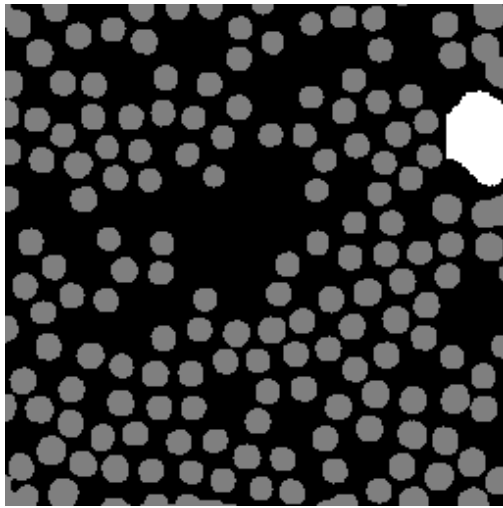


Figure 74: Output image of the final model trained with augmented data set III visualised with Figure 64 as input image.