

# Identification of Martensite Bands in Dual-Phase Steels: A Deep Learning Object Detection Approach Using Faster Region-Based-Convolutional Neural Network

Niklas Fehleemann,\* Ana Lia Suarez Aguilera, Stefan Sandfeld, Felix Bexter, Maximilian Neite, David Lenz, Markus Könemann, and Sebastian Münstermann

Martensite banding in dual-phase steels is an important research topic in the field of materials design, since it affects the local damage properties of the material largely. Therefore, it is necessary to quantify the amount and the geometrical details of the bands in a specific microstructure, for example, for simulative approaches. A convolutional neural network is trained on manually labeled scanning electron microscopy images of DP800 steel and a subsequent effort is made to transfer these results to statistical quantities for the generation of representative volume elements (RVE). As exact geometric definitions of martensite bands in 2D are difficult, the influence of different band definitions is investigated. The result of the training shows good prediction accuracy but is strongly dependent on the chosen band definition and the underlying human bias from the labeling process. A statistical analysis using cross-validation shows that reliable results can already be achieved with only small datasets of around 50–100 training images due to the transfer learning approach. This is an important outcome as it eliminates the need to generate a large dataset which can only be obtained from time-consuming microscopy work and manual labeling of the images.

research field of process–(micro)structure–property–performance relations. Increasingly, machine learning methods are also being used here, for example, to accelerate material design,<sup>[1]</sup> to link grain sizes and mechanical properties,<sup>[2]</sup> or to describe dislocation networks.<sup>[3]</sup> The specific microstructure has a big influence on basic parameters such as the toughness or the yield stress but even more on local properties on the microscale like the damage initiation and accumulation<sup>[4]</sup> or the fatigue properties.<sup>[5]</sup> For well-known steels such as dual-phase (DP) steels, the relations between microstructure and damage properties have already been studied in a vast number of publications, see, for example, Pütz et al.<sup>[6]</sup> for investigations about the damage tolerance of DP800 and DP1000, Tasan et al.<sup>[7]</sup> regarding simulative experimental studies about stress and strain partitioning in DP800, or Heibel et al.<sup>[8]</sup> for studies on the edge crack sensitivity of different DP

## 1. Introduction


The drive for evermore efficient construction, design and layout of components, buildings, and vehicles also affects materials science to a large extent. For some time now, research has focused on the adjustment and tailoring of microstructure in the emerging

and complex phase steels. Experimentally, however, it is almost impossible to quantify the specific effects of individual microstructure parameters (e.g., grain size or phase ratio), since the modification of a particular parameter normally also modifies some or all of the other parameters.<sup>[9]</sup> In order to overcome this problem, simulation-based approaches using virtual representations of the microstructure are suitable. A commonly used tool for this kind of analysis is the so called representative volume element (RVE). These are sections of the volume, whose specific properties are of a nature that is representative of the behavior of the aggregate material. For this, the following scale separation must apply:  $L_{\text{micro}} \ll L_{\text{rve}} \ll L_{\text{macro}}$ .<sup>[10]</sup> Since the microscopic characteristic length depends strongly on the material, the required size of the RVE also depends on the properties of the material.<sup>[11]</sup>

RVE are distinguished from real structure models of microstructure. These kinds of models can be generated directly from microstructure images obtained, for example, from light optical microscopy (LOM) or electron backscatter diffraction (EBSD). They are for instance used by Tasan et al.<sup>[12]</sup> In this case, the LOM/EBSD image is directly used as the 2D model to simulate the underlying mechanical properties of the experimentally observed microstructure. In contrast to that, the different parameters of the microstructure (grain size, orientations, phase ratios) can be represented by some kinds of distribution functions

N. Fehleemann, A. L. Suarez Aguilera, F. Bexter, M. Neite, D. Lenz, M. Könemann, S. Münstermann  
Integrity of Materials and Structures  
RWTH Aachen  
Intzestrass 1, 52072 Aachen, Germany  
E-mail: niklas.fehleemann@ieh.rwth-aachen.de

S. Sandfeld  
Institute for Advanced Simulation  
Materials Data Science and Informatics (IAS-9)  
FZ Jülich  
Wilhelm-Johnen Strasse, 52428 Jülich, Germany

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/srin.202200836>.

© 2023 The Authors. Steel Research International published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/srin.202200836

which are then used to generate input to build so-called statistically RVEs (S-RVE), which rely on some kinds of averaged quantities (e.g., used by Liu et al.<sup>[13]</sup> and Vajaragupta et al.<sup>[14]</sup>). While models build from the real microstructure might be slightly more precise in displaying the exact morphology of the constituents, a statistical description of the microstructure makes it easier to vary specific features of the microstructure. This is inevitable for the use of RVEs for microstructure design and the quantification of the effects of single constituents of the microstructure.

Various open-source software packages can be used to generate RVE (so-called RVE generator). These include DREAM.3D,<sup>[15]</sup> Neper,<sup>[16]</sup> or Kanapy,<sup>[17]</sup> utilizing different approaches to build the microstructure model and map different properties. Recently, an RVE generator has been developed in the authors' group, which uses discrete representations of ellipsoids to represent grains. For more details on this, the reader is referred to Henrich et al.<sup>[18]</sup>. In this approach, the focus was put on the correct representation of the geometrical description of the microstructural constituents. Therefore, ellipsoids were chosen as a mapping for the grains, which are generally better suited as a representation for the grains than spheres, since geometrically preferred directions (e.g., due to rolling processes) can be better described. In order to map the distributions of the necessary grain parameters (e.g., semiaxes of the ellipsoids and inclination angle of the ellipsoids), a log-normal or Gamma distribution can be fit to the individual data series.<sup>[19]</sup> However, previous studies have shown that there are interdependencies between the individual grain parameters that cannot be represented by univariate distribution functions. Therefore, a so-called "Wasserstein generative adversarial network" is used in the RVE generator, which is able to serve as a multivariate distribution function for synthetic grains. More details about this approach have been described in previous studies.<sup>[20,21]</sup> This approach is suitable for ellipsoidal or sphere-like structures such as ferrite grains, martensite islands, voids, inclusions, etc., but not for more complicated morphologies such as martensite or pearlite bands.

Martensite banding, especially in DP steels, is an active research topic, since bands of a secondary hard phase lead to numerous changes in the local properties, most of which influence the local properties in a negative way.<sup>[22,23]</sup> However, to our knowledge, there is no method to describe the individual bands of a microstructure quantitatively, while taking into account the geometric features of the bands. While there are some approaches to quantify the amount of banding in a microstructure,<sup>[24,25]</sup> this description is not sufficient for the representation in S-RVE: Only if bands and islands can be described independently from each other and the rest of the microstructure, the individual effect of the bands on the microstructure and the mechanical properties can be analyzed. However, this leads to the question of how exactly bands and other representations (e.g., martensite in the form of islands) of the same phase can be distinguished. In dual-phase steels, where the microstructure has a generally elongated structure due to the rolling process, this question is not trivial to answer and requires time-consuming and manual classifications by experts for the respective material.

One way to accelerate such a problem is to use machine learning methods, especially from the field of supervised learning.

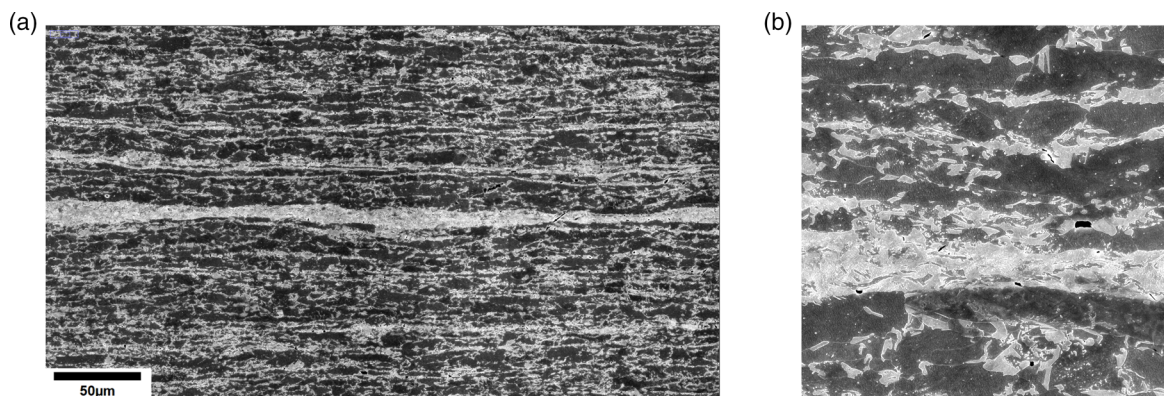
Supervised learning is already widely used and successfully applied in the field of material characterization, for example, for the classification of damage mechanisms,<sup>[26,27]</sup> grain structures,<sup>[28]</sup> or microstructural constituents.<sup>[29,30]</sup> The advantage is that after experts have manually created a dataset for training, the model can automatically process new data without human intervention.

In this study, we present a machine learning approach for automatic detection and localization of martensite bands in scanning electron microscopy (SEM) images from a DP800 and the ensuing transfer to statistical quantities for RVE generation. For this purpose, a state-of-the-art object detection algorithm, the Faster region-based-convolutional neural network (CNN), is implemented and used. We show that this algorithm is able to learn and subsequently apply different definitions of martensite bands even when only relatively few data is available. The article is structured as follows. Section 2 describes two different strategies to separate the bands from the islands (so-called labeling) to generate training data. In Section 3 the used network architecture is explained. In Section 4, the results of the two different definitions are presented. Section 5 contains a discussion and comparison of the results. In Section 6, we show subsequently how the detected bands are transferred into a statistical representation for the use in RVE and which different peculiarities result from the varying definitions. Thus, we provide an end-to-end approach for data generation for RVE, ranging from the band detection to the statistical representation.

## 2. Labeling Approach for Martensite Bands

The material examined in this study is a commercial dual-phase steel DP800. The material has a phase fraction of martensite of around 32% and shows a significant martensite banding of the microstructure (oriented along the rolling direction RD)). Other phases such as bainite play only a very minor role for important properties of this specific steel and are therefore omitted from this study. The yield stress is 460 MPa, the ultimate tensile strength has a value of around 823 MPa, and the fracture strain is at 21%. There is some scatter in the fracture strain for tensile loading, with fracture strains at 19.84, 19.6, and 23.5% for three tensile tests. In a previous study, the pronounced banding was found responsible for this scatter in the fracture strain.<sup>[6]</sup> Figure 1a shows a panoramic SEM image of the microstructure with a height of 320  $\mu\text{m}$  and a width of 1153  $\mu\text{m}$ . The data used in this study was originally generated for another study, where the effect of local strains and stress triaxiality on the void nucleation and evolution of this DP800 was investigated.<sup>[31]</sup> The image was taken by means of a SEM-Inlens-Detector, which allows a very clear delimitation of martensite and ferrite. The image was originally taken as a combination of several 100  $\mu\text{m} \times 100 \mu\text{m}$  images, which were then stitched together using an algorithm. The resolution of the original images corresponds to 3072 pixels per 100  $\mu\text{m}$ .<sup>[31]</sup>

The x-axis is oriented along the RD and the y-axis along the sheet normal (RD  $\times$  SN). A closer look on an exemplary 50  $\mu\text{m} \times 50 \mu\text{m}$  excerpt (Figure 1b) from the panoramic SEM image reveals that the banding and the clear preferred direction along the RD comes from elongated islands and more



**Figure 1.** SEM images of DP800 steels. a) SEM-Inlens image of DP800 steel. Martensite shown in white, ferrite in black. The size is 1153  $\mu\text{m}$   $\times$  320  $\mu\text{m}$ . b) 50  $\mu\text{m}$   $\times$  50  $\mu\text{m}$  excerpt from the microstructure.

pronounced bands with partially blocky dimensions, extending over a larger area (in this example over the whole 50  $\mu\text{m}$ ).

Images taken at 90° to those shown in Figure 1 (transverse direction  $\times$  SN) show a very similar microstructure. From this, it can be concluded that the (larger) bands are more likely to be 3D “plates”, which have a greater extension in the RD and in the transverse direction.

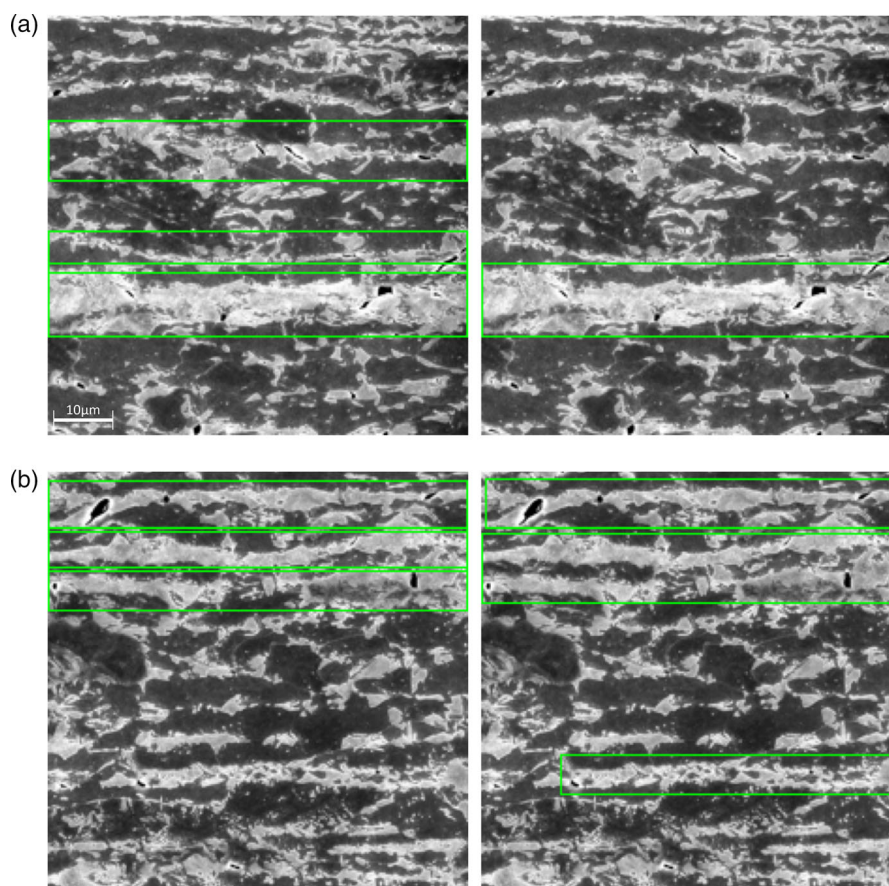
For the construction of S-RVE, the statistics of the elongated islands are captured using ellipses or ellipsoids, but for the blocky bands/plates case these are unsuitable. The representation of a band as a centered oblate ellipsoid in the RVE cannot accurately represent the geometric shape of a band/plate, for example, enclosed ferrite grains cannot be represented in this way. In addition, a large ellipsoid would lead to a decrease in the thickness of the band from the center of the RVE to the edges. Therefore, the bands must first be separated from the island in order to then be displayed in the RVE. Thus, it is necessary to define what is a band and what therefore needs a special treatment in the RVE generation process and what can be represented with the standard ellipsoids. To achieve this, different definitions were made to label the bands according to different criteria. With this, training data for the object detection approach described in the next section is generated. To our knowledge, there is neither a benchmark nor “right” or “wrong” to the question “what is a martensite band?” In general, the answer to this question depends on the goal of the investigations (i.e., the accurate representation of the martensitic morphology in S-RVE and the investigation of the influence of the bands on the damage behavior).

Hence, this study examines two different definitions of bands and tests and discusses them in terms of usability. The goal is to find out whether an object detection approach is capable of handling different definitions of martensite bands. The definitions are applied to 50  $\mu\text{m}$   $\times$  50  $\mu\text{m}$  excerpts of the microstructure, with the size chosen accordingly to the edge length of the RVE, which has edge lengths in the range of 30–60  $\mu\text{m}$ . During the labeling process we mark an area as a band by drawing a rectangle (so called bounding box) around the area, enclosing the band. This was done using the “Image Labeler app” from the “Image Processing and Computer Vision Toolbox” in MATLAB.<sup>[32]</sup>

The following two definitions were used to label all images: 1) Definition 1 (**Def 1**): A band is an elongated area of martensite that extends from the left side of the image to the right side of the image. The bands should have no or very small breaks ( $\leq 10$  pixels). Usually, the bands are parallel to the  $x$ -axis, but slight deviations (5°) are allowed. The height of the band does not need to be constant throughout the length, so the band may be very thin at some points. Bands also include blocks of martensite that have enclosed areas of ferrite, as well as curved or branched sequences. 2) Definition 2 (**Def 2**): As opposed to Def 1, bands can be shorter than the width of the image. Breaks of the band are allowed if the band-like structure remains “recognizable” for a human with the respective domain knowledge and experience. Contrary to the first definition, the bands should have an almost constant height and be parallel to the  $x$ -axis. The bands must be different from the rest of martensite islands surrounding the band. Very thin band-like structures are not labeled as bands.

These definitions are no strict, mathematical definitions but are rather intended as guidelines for a human expert for manual labeling of martensite bands. In doing so, a certain degree of human bias will automatically be introduced into the data. The effects of this will be discussed in Section 5. Additionally, both definitions implicitly assume that all images have the same size and are not scaled. **Figure 2** shows the resulting bounding boxes for both definitions. Here, it can be seen that Def 1 focuses on the coherence of the martensite structure across the entire image width and all associated side arms. In contrast, Def 2 focuses more on the blockiness of the martensite. The constant thickness is more important than the length and possible side arms or branched sequences. In both definitions, the martensite content in the band area is significantly increased compared to the average value of the material. According to the definitions, the two structures above the big band at  $\approx 1/3$  of the height are labeled as bands according to Def 1 because they span over the whole image, but not according to Def 2, since they do not show a constant height and are relatively thin.

The panoramic SEM-Inlens (Figure 1) image is split into 138 excerpts. For Def 1, 97 bands were labeled in 69 of the 138 images. For Def 2, 51 bands were labeled in 40 of the 138 images,



**Figure 2.** Exemplary images labeled according to Def 1 and Def 2. a) Comparison between Def 1 (left) and Def 2 (right). The two structures above the pronounced bands do not show a complete horizontal alignment and are therefore not labeled according to Def 2. b) The comparison between Def 1 (left) and Def 2 (right). The banded structure on the lower half of the image does not stretch over the whole image and is therefore not labeled according to Def 1.

the remaining images without bands are not used. These data is used as the ground truth. The nearly doubled number of instances in Def. 1 as compared to Def 2 can indicate that Def 2 is a more strict definition.

In addition to the SEM-Inlens images, labeling experiments were also performed on image from a standard SE detector. The problem arises that on these images martensite and ferrite are only well distinguishable on the basis of the grain boundaries and there is no clear color contrast. This complicates both the labeling and the automatic binarization, which is important for the separation of martensite and ferrite for the generation of RVE input data. Therefore, we stick to the Inlens images for all further analysis.

### 3. Object Detection

#### 3.1. Theoretical Foundations

This article deals with the localization and classification of objects in an image (even though the classification is somewhat needless in our case, since there are only bands and no other components to be recognized). There are two main ways to do this: image segmentation and object detection. Currently, all state-of-the-art algorithms of both model classes are based on

deep learning (DL) methods when processing images or other grid-like data (e.g., videos or time series, a special form of a multilayer neural network is used, the so-called CNN). For a detailed introduction into CNNs and the convolutional operation, the reader is referred to Goodfellow et al.<sup>[33]</sup> For image segmentation and object detection tasks, CNNs show superior results as compared to many classical image analysis algorithms and recipes, making them the basis for most of the state-of-the-art image analysis methods.

Image segmentation for a binary classification task combines the pixels belonging to an object in an image or a volume, segmenting them from the background. A well-known network architecture for such tasks is the U-Net, developed by Ronneberger et al.<sup>[34]</sup> In contrast to that, object detection deals with the localization and subsequent classification of objects in an image by drawing a bounding box around the object.<sup>[35]</sup> A disadvantage of object detection is that the exact shape of the object cannot be detected. In this work, however, an object detection approach was chosen, since the exact shape of the bands is not crucial for the representation in the RVE (e.g., due to lower resolutions of the RVE compared to the image), and complete masks with all shapes introduce additional, unwanted complexity into the labeling problem.

For object detection algorithms, a distinction is made between two-stage models, which have separate models for extracting the so-called region proposals (RPs) (regions in an image where objects may or may not lie) and classifying the objects and second one-stage models, which skip the RPs.<sup>[35]</sup> Popular one-stage architectures are presented in the study by Redmon et al.<sup>[36]</sup> and Lin et al.<sup>[37]</sup>. The benefit of one-stage models is the simpler architecture and the faster prediction speed.<sup>[37]</sup> Since the prediction speed is not crucial for the detection of microstructural features (since, unlike e.g., autonomous driving, this is not a real-time application), a two-stage architecture was chosen.

A popular family of two-stage algorithms is the R-CNN model family, where R-CNN stands for “region-based convolutional neural networks.” The family consists of three subsequent models: the R-CNN from 2014,<sup>[38]</sup> the Fast R-CNN,<sup>[39]</sup> and the Faster R-CNN,<sup>[40]</sup> both of which were published in 2015.

The initial architecture is the R-CNN, proposed by Girshick et al.<sup>[38]</sup>. This architecture is divided into three parts. In the first step, a selective search (SS) algorithm is used to generate 2000 RP for each image. A RP is a location from the image, where an object may or may not lie. These proposals are resized to squares and subsequently fed to a CNN, which serves as a feature extractor and gives a feature vector with 4096 entries to a support vector machine algorithm, which classifies the objects and calculates the bounding box (given by the four corner coordinates). However, different problems arise from this architecture: At first, it is extremely inefficient in terms of time and storage space. Since features are computed and stored for each RP, the R-CNN with very deep CNN architectures as a backbone (up to several hundred layers) takes a very long time for training and predictions. In addition, several hundred gigabytes of memory are required.<sup>[39]</sup> Furthermore, the training process consists of several steps (CNN training, support vector machine training for classification, and support vector machine training for bounding box regression), which are also inefficient and prohibit training in one single loop.<sup>[39]</sup>

To deal with this drawbacks, the Fast R-CNN was developed. With this, the time and memory problem is contained by no longer computing a feature map for each of the 2000 RPs, but only once for the entire image. Then the RPs are computed directly from the feature map. This saves both memory and time by reducing unnecessary computations.<sup>[39]</sup>

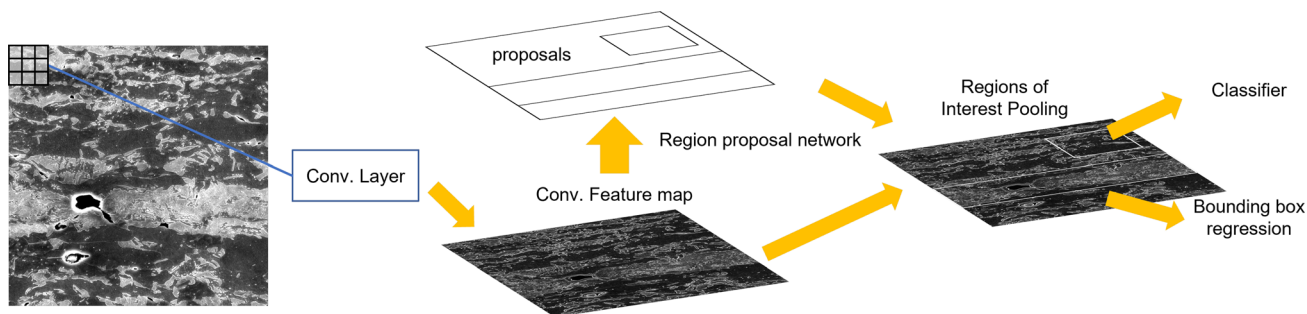
The next design improvement, the Faster R-CNN (Figure 3), eliminates the last bottleneck, which is the generation of RPs using SS. This is illustrated by the fact that, if the time spent

for the generation of RPs is ignored in Fast R-CNN, almost real-time object detection is already possible.<sup>[40]</sup> To get closer to true real-time object detection, the SS is replaced by the so-called RP network in the Faster R-CNN. This RP network shares the convolutional layer with the detection network, making the generation of RPs very efficient.<sup>[40]</sup> In Figure 3 it becomes clear that there is only one single fully convolutional layer in the Faster R-CNN. This CNN with shared computational layers can be trained with the so-called approximate joint training. This training style can speed up the training by up to 50%, compared to other training styles.<sup>[40]</sup>

The Faster R-CNN is already being used in materials science, for example, for the detection of defects on the surface of steel sheets<sup>[41]</sup> or in weld seams.<sup>[42]</sup> However, these papers are more concerned with technical modifications to the structure of the Faster R-CNN and not with the effects of different labeling approaches on the statistical properties of the microstructural components to be detected. Here, an effort is made to use the Faster R-CNN in a setting where the properties to detect are not exactly defined.

### 3.2. Implementation Details

For this study, the Faster R-CNN is implemented in python using PyTorch version 1.11 and torchvision version 0.12.<sup>[43]</sup> With the PyTorch implementation of the model, we use a pretrained version of the network with a ResNet50-backbone (for more details on the ResNet neural network architecture for computer vision, the reader is referred to He et al.<sup>[44]</sup>). This significantly speeds up training and allows good predictions to be made on relatively few images. This is an advantage in that the generation of good quality SEM or EBSD images with reproducible properties in high quantities is much more time-consuming than, for example, the generation of a large quantity of simple finite-element simulations. The concept of only fine tuning the weights of the most specialized layers of a pretrained model is called transfer learning and additionally accelerated our training process significantly. During the training, the model predicts bounding boxes with a so-called confidence score. The bounding boxes are recorded if the confidence score is higher than 0.8. All bounding boxes are stored in the format  $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$  and the images were resized from  $1600 \times 1600$  pixels to  $512 \times 512$  pixels. The smaller size increases the training speed and prevents problems with the RAM during training. See Section 4 for a comparison of the results of different image sizes.



**Figure 3.** Schematic layout of the Faster R-CNN.

As introduced in Section 2, two definitions were evaluated in this article. For both, the detector was trained for 50 epochs. This training was run using a NVIDIA V100-SXM2 GPU provided by the RWTH High Performance Computing Cluster and takes  $\approx 75$  min for the 69 images of Def 1 and 45 min for the 40 images of Def 2. To improve the small size of the training dataset, data augmentation techniques were applied to the images. For this, the Albumentations-Library<sup>[45]</sup> was used. We evaluated different augmentation techniques and list the best augmentations below ( $p = x\%$  denotes the percentage of the training instance to which the augmentation was applied): 1) Flipping ( $p = 20\%$ ): flips image vertically, horizontally, or in both ways; 2) Affine transformation ( $p = 50\%$ ): applies translation, shearing, and rotation to the image (in defined boundaries); 3) MotionBlur ( $p = 20\%$ ): applies motion blurring to the image; and 4) Blur ( $p = 10\%$ ): blurs image with random sized kernel

The choice of augmentation methods and respective parameter ranges resulted from the fact that the microscopy images were mainly aligned such that bands are always horizontal. Therefore, rotational transformation of the images with arbitrary angles is not physically reasonable and was excluded.

The original training dataset contains 69 (Def 1) and 51 bands (Def 2), which is, by comparison with training datasets, typically used in computer science applications an extremely small number. In order to check whether the results strongly depend on the selected train-test split and whether overfitting might take place, a fivefold cross-validation was performed. In a  $k$ -fold cross-validation, the dataset is divided into  $k$  equally large subsets. Each of these  $k$  parts is then used once as a test dataset, while the remaining  $k - 1$  parts are used as training data. Therefore, in a  $k$ -fold cross-validation,  $k$  trainings are performed and then statistically analyzed. Due to the pretrained backbone, this is possible in a reasonable time.

Note that cross-validation is only used to detect certain problems in the training data; the final model is then trained and evaluated separately based on all available data.

### 3.3. Evaluation Metrics

To quantify the success of the trained models, the results are evaluated based on the “recall” and “precision” metrics. These are defined as follows

$$\text{Recall} = \frac{T^+}{T^+ + F^-} \quad (1)$$

$$\text{Precision} = \frac{T^+}{T^+ + F^+} \quad (2)$$

where  $T^+$  means true positive (band correctly located and classified as band),  $F^+$  is false positive (fraction of the image incorrectly defined as band by the model), and  $F^-$  is the false negative (missed to identify image fraction as band). Another metric that is often used with the recall is the specificity  $\frac{T^-}{T^- + F^+}$ . However, this cannot be clearly calculated here as there is no definition for true negatives. Theoretically, any island or martensite clustering that is correctly identified as background (everything that is not martensite band counts as background in this work, i.e., ferrite matrix, martensite islands, and possible

pores) is considered a true negative, but it is difficult to draw the line here, so this value is not used in this study. A prediction is assumed to be true positive if the predicted and the ground truth bounding boxes match to at least some fractional value. For this purpose, the so-called Intersection-over-Union (IoU) threshold is used. It is defined according to the Equation (3).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (3)$$

In this equation, the Area of Overlap is the common area between the predicted and the ground truth bounding box. The Area of Union is the area enclosed by both boxes together. In general, a true positive is recorded if the IoU is greater than 0.5. In this study, the detectors are evaluated at IoU scores of 0.5 and 0.75. The precision and recall scores are calculated for both thresholds and the mean value over test set is used to evaluate the detector. As an additional metric, we introduce the “accuracy”, which is simply the mean of recall and precision.

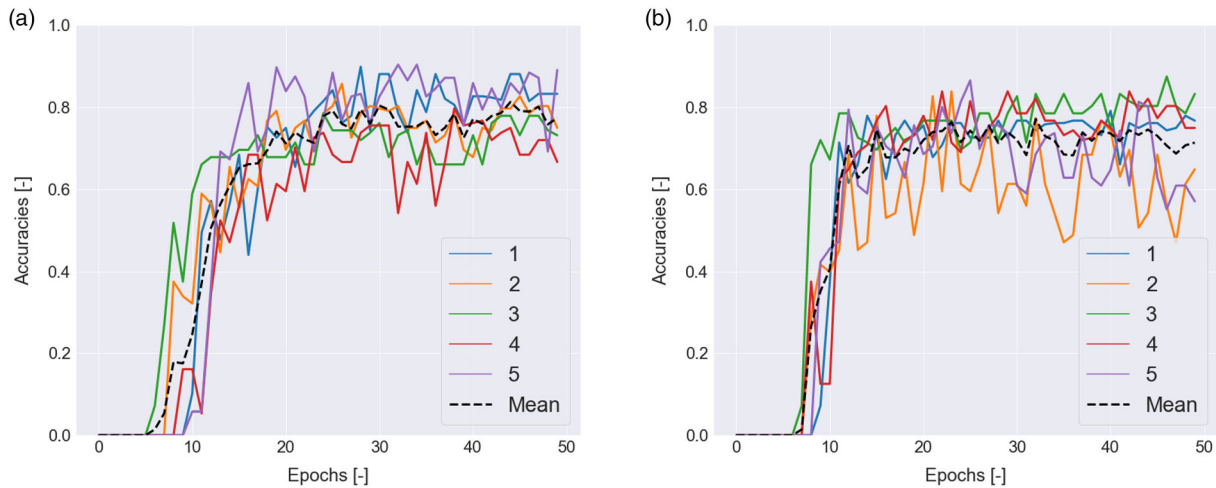
## 4. Model Results

### 4.1. Results of the Cross-Validation for Def 1

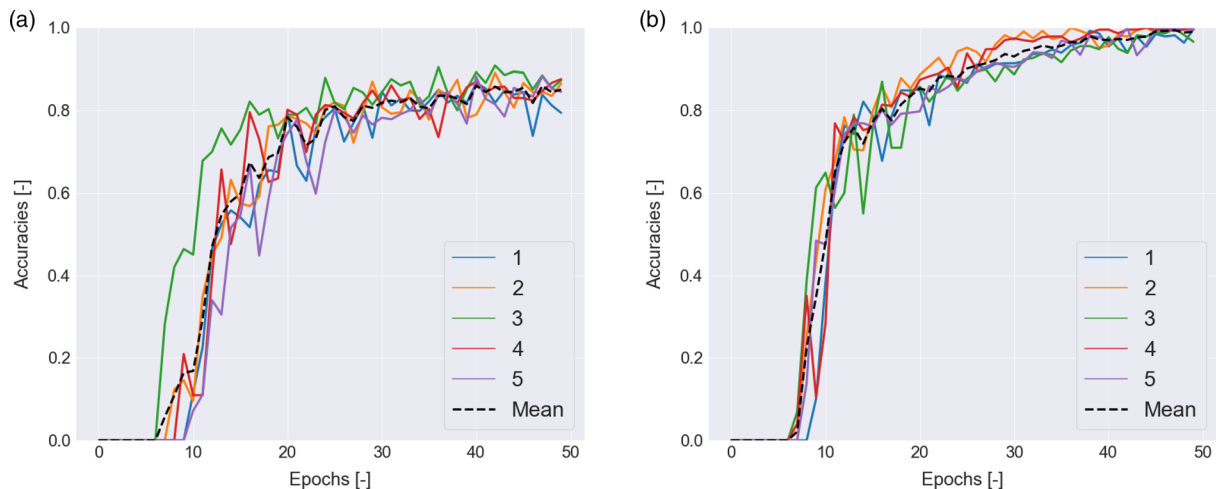
The results for the fivefold cross-validation using training instances according to Def 1 are shown in Figure 4. Here, the accuracy (see Section 3.3) is shown for the respective validation part of the dataset, without data augmentation (right part, Figure 4b) and with the data augmentation (left part, Figure 4a). The different lines indicate the different folds of the cross-validation and the mean of all folds is shown with the dashed black line. It becomes clear that —although there is some scatter in the curves—all five folds, that is, all validation dataset, behave approximately in the same way. The shapes of the curves, as well as the final accuracy level, are approximately similar for both the data with augmentation and without augmentations. The only outliers are two folds for the training without data augmentations, which tend to fall again during the training process. This indicates also that the dataset without data augmentation are statistically not (sufficiently) representative; hence, data augmentation is important in this case.

Slightly different results can be seen for the training parts of the dataset (Figure 5). When comparing the results without data augmentation (right side of Figure 4 and 5), it becomes clear that without data augmentation the dataset is too small such that the trained model suffers from overfitting. While the training curves increase constantly and are almost all at 100% accuracy from epoch 40 on, the validation curves are only at around 75% accuracy, with the already described drop for two particular folds. For the training with data augmentation (left part of Figure 4 and 5), no such behavior is evident. All curve reach values at around 80% accuracy, with slightly less oscillations in the training curves, due to the higher number of samples.

Table 1 shows the summarized results for the cross-validation. For a more complete analysis, we also analyzed the effect of different image resolutions on the prediction quality: The chosen sizes were  $100 \times 100$  (resulting in a resolution of 2, meaning that  $1 \mu\text{m}$  is equal to 2 pixel),  $512 \times 512$  (the standard size), and



**Figure 4.** Accuracies of predictions for the validation datasets during cross-validation for Def 1. The different lines with labels 1–5 indicate the five different folds of the cross-validation, the dashed black line is the mean computed over all five folds. a) Accuracies using data augmentation. b) Accuracies without data augmentation.



**Figure 5.** Accuracies of predictions for the training datasets during cross-validation for Def 1. The different lines with labels 1–5 indicate the five different folds of the cross-validation, the dashed black line is the mean computed over all five folds. a) Accuracies using data augmentation. b) Accuracies without data augmentation.

**Table 1.** Effect of different image resolutions and sizes on the cross-validation results. The mean over the five folds is shown for the validation set. All values in percent, image side length in pixel.

Image side length [px]	Precision	Recall	Accuracy
100	72.78	72.29	72.54
512	79.23	75.60	77.75
1024	71.03	78.64	74.84

$1024 \times 1024$ . From the results it can be deduced that even a very low image resolution of only 2 leads to good results. The image size of  $512 \times 512$  has the best precision and accuracy, while the image size of 1024 has the highest Recall. These results show that an image size of  $512 \times 512$  pixels has the highest values

in the overall accuracy, so this size is used for further analysis, although the differences are not serious.

#### 4.2. Results of the Cross-Validation for Def 2

As in the previous section, the results of the cross-validation are shown for Def 2. These can be seen in Figure 7. In this figure, comparable results to the results for Def 1 can be seen. There is only minor scatter between the different folds and no unexpected progressions in the curves can be observed. For the results with respect to the not-augmented data (Figure 7b), there is a sharp drop of the accuracy for one of the folds down to 60% accuracy. From both of the curves, it becomes clear that the results are better on average than the results of Def 1, with accuracy values oscillating more around 90% than 80%.

As described in the previous section, the model for Def 1 shows problems with overfitting. While comparing the accuracy values for the validation and training dataset, without data augmentation for Def 2 (Figure 6b and 7b), a similar behavior is observed, with the values for the training sets being extremely close to 100% from epoch 30 on. Although, the difference between the validation and the training values is not that pronounced compared to Def 1, it indicates less problematic overfitting for Def 2.

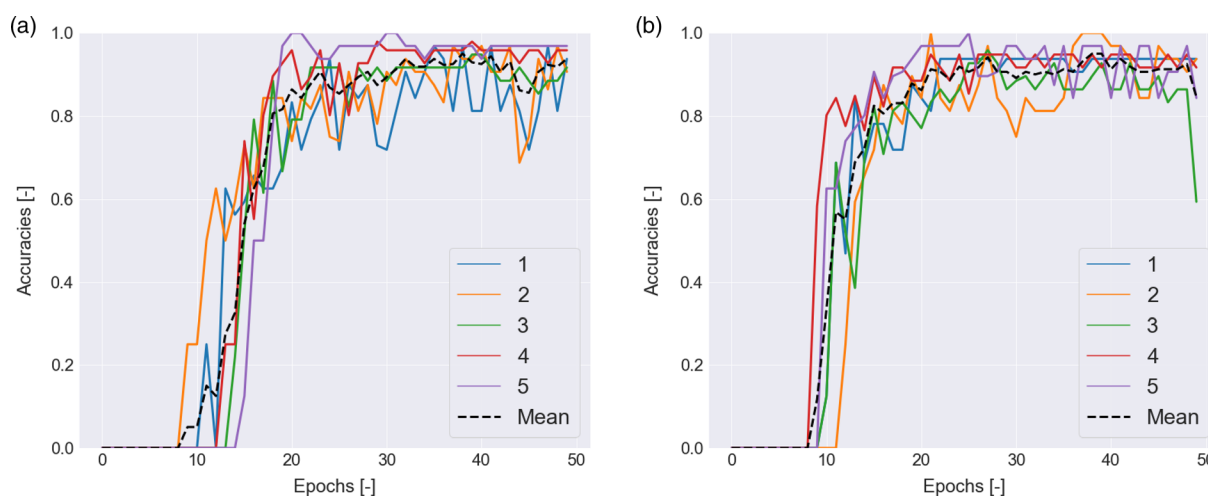
Table 2 shows the mean results for the cross validation of Def 2. Similar results as for Def 1 can be observed. Again, the accuracy is highest at an image size of  $512 \times 512$ , but the differences are very small (less than one percentage point). Overall, the results shown in Table 1 and 2 confirm the basic assumption of an image size of  $512 \times 512$  as good, there is no other image size that clearly produces better results.

**Table 2.** Effect of different image resolutions and sizes on the cross-validation results for Def 2. The mean over the five folds is shown for the validation set. All values in percent, image side length in pixel.

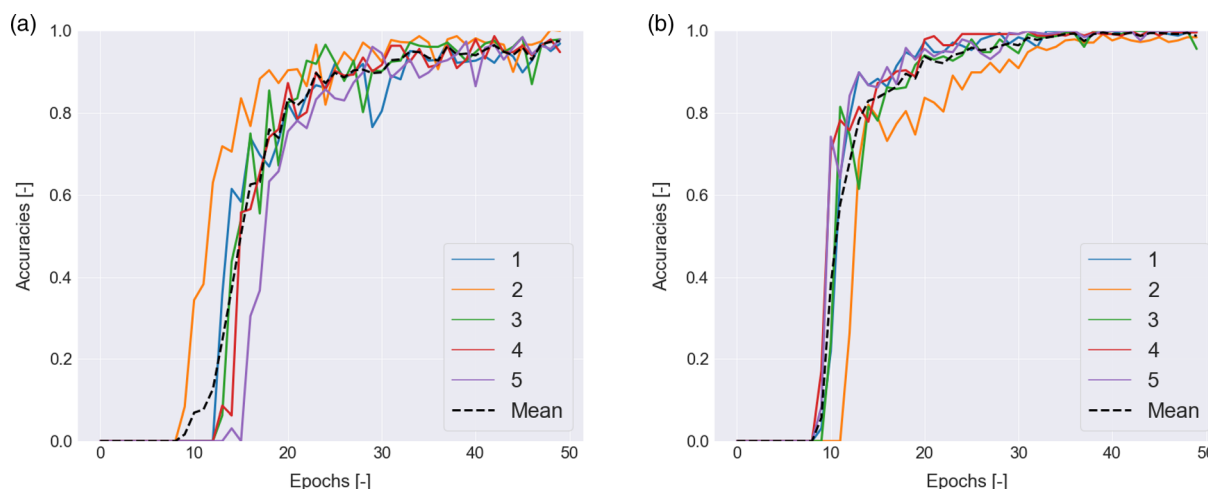
Image side length	Precision	Recall	Accuracy
100	91.25	93.33	92.29
512	93.75	93.75	93.75
1024	95.00	90.00	92.25

### 4.3. Results for the Final Model (Def 1 and 2)

Due to the evidence of overfitting, we discarded the results without data augmentations and consider only the results with data augmentation for a more detailed analysis of both different definitions.



**Figure 6.** Accuracies of predictions for the validation datasets during cross-validation for Def 2. The different lines with labels 1–5 indicate the five different folds of the cross-validation, the dashed black line is the mean computed over all five folds. a) Accuracies using data augmentation. b) Accuracies without data augmentation.



**Figure 7.** Accuracies of predictions for the training datasets during cross-validation for Def 2. The different lines with labels 1–5 indicate the five different folds of the cross-validation, the dashed black line is the mean computed over all five folds. a) Accuracies using data augmentation. b) Accuracies without data augmentation.

For the final model, the respective dataset is split into a 75/25% train test split. The resulting metrics for Def 1 are shown in **Table 3**. The results are calculated for IoU threshold of 0.5 and 0.75, respectively. The accuracy values are in good agreement with the last values from the curves from the cross-validation in Figure 4a, indicating no unexpected abnormal behavior in the final model.

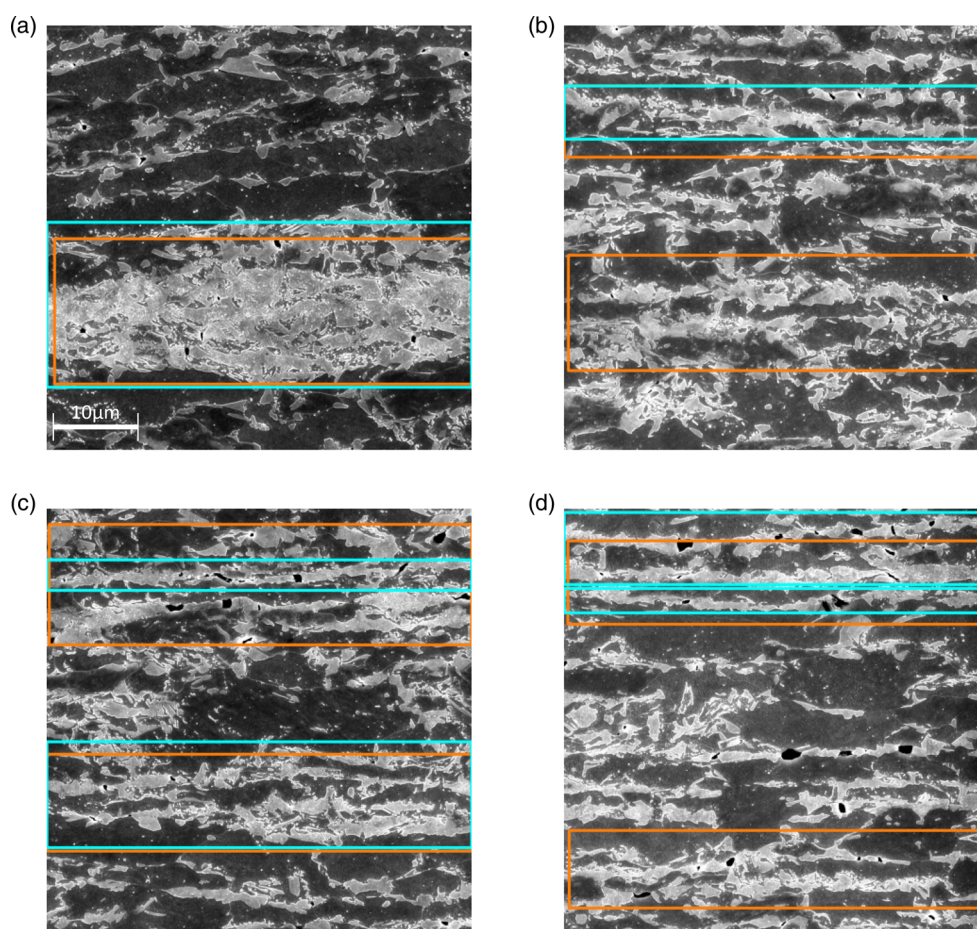
Both rows of Table 3 show that the recall for this model is higher than the precision, about 12 percentage points for both thresholds. This indicates that the model generates too many false positives for this dataset but recognizes the majority of the correct bands (at least for IoU threshold of 0.5). Thus, the recall is relatively high. For the IoU threshold of 0.75, similar

**Table 3.** Evaluation metrics for the final model for Def 1. The metrics are calculated from the final model w.r.t. to the test set. All values in percent.

Threshold	Precision	Recall	Accuracy
0.5	68.52	80.56	74.54
0.75	54.63	66.67	60.65

results are observed, with all three metrics around 14 percentage points lower than for an IoU threshold of 0.5. The precision of only around 55% indicates that there are nearly as many false positives than true positives at this threshold.

In **Figure 8**, some examples from the test dataset for Def 1 are shown. The ground truth bounding boxes are drawn in light blue, while the predicted boxes are drawn in orange. For a very pronounced banded structure and only small martensite islands like in Figure 8a, the prediction is almost perfect. All metrics are 1, even at a higher IoU threshold of 0.75. For excerpts that are a bit more indistinct in terms of the martensite morphology (Figure 8b–d), the prediction quality differs: For the excerpt in Figure 8b, the band in the upper half of the image is correctly identified (At IoU 0.5, at IoU 0.75, the difference is to big). In addition to that, a big bounding box is predicted in the lower half of the image, enclosing a banded structure, which was not labeled according to Def 1, because it did not span completely over the whole image. For the excerpts in Figure 8c,d, it is clear that the bounding boxes at the top of the images are basically correctly placed, but incorrectly sized. In Figure 8c, the prediction is much bigger than the real bounding box, enclosing more



**Figure 8.** Exemplary results for Def 1. The predicted bounding boxes are colored in orange and the ground truth boxes in blue. Metrics are denoted for each picture in the subcaptions for an IoU threshold of 0.5. a) Very pronounced band is detected without problems (Prec=1, Rec=1, Acc=1). b) False positive in the lower half of the image (Prec=0.5, Rec=1, Acc=0.75). c) Bounding box predicted at the right location (upper side of the image) but with wrong dimensions (Prec=0.5, Rec=0.5, Acc=0.5). d) Bounding box predicted at the right location (upper side of the image) but with wrong dimensions (Prec=0.0, Rec=0.0, Acc=0.0).

of the morphology in the upper 1/3 of the image than that which was labeled. In Figure 8d, the two ground truth boxes are wrongly divided by the prediction, leading to precision and recall of 0.0 in this excerpt. Additionally, there is a false positive in this excerpt, which is, similar to 8b, a banded structure which does not range over the whole image.

In order to have a better comparability with the results of Def 1, we also trained the final mode for Def 2 with data augmentation. This is supported by the fact that for Def 2 the results without data augmentation show slight signs of undesirable behavior, for example, the aforementioned sharp drop for one of the folds. For the final model, a 75/25% train test split is also applied. The results are displayed in **Table 4**. It becomes clear that the metrics are overall better for Def 2 than for Def 1 and the model is able to produce very accurate predictions for the test dataset. The results are also well in line with the cross-validation curves in Figure 6a, where the individual folds are in the 90–100% range at epoch 50. With the high values for all of the three metrics, we chose to omit, showing exemplary images for this definition. For both thresholds, only in one image the predictions are incorrect, in all others all metrics are at 100%

## 5. Comparison and Discussion

The results show that our proposed object detection approach is capable of separating martensite bands from the from the rest of the martensitic morphology in a good manner, with accuracy values around 75–100%. Also, the cross-validation shows that the number of samples in our datasets seems to be sufficient and there is no evidence for bigger problems with the scarcity of the data (which would, for example, be indicated by extremely differing curves for the folds in Figure 4 and 6). This is a big advantage of the chosen transfer learning procedure (using a pre-trained ResNet in the Faster R-CNN), enabling us to work with a relatively small amount of data, which is enhanced by data augmentation techniques. Given that images from costly electron microscopy are needed to distinguish martensite and ferrite properly, the small amount of data required is a clear advantage of our approach.

However, the prediction quality differs strongly between the two definitions, even with the same configuration of hyperparameters, like the chosen augmentations, epochs, batch size, etc. While Def 2 is nearly perfectly mapped at around 100% with no big differences between precision and recall, there is only a prediction accuracy of around 75% of the model with respect to Def 1. Also, this model shows a skew in the metrics toward recall over precision, indicating more false positives than false negatives. Additionally, the dropoff in the prediction metrics from an IoU threshold of 0.5 to one of 0.75 is higher for Def 1 (around 14 percentage points for all three metrics) than for Def 2 (10

percentage points for precision and 5 percentage points for recall). This shows overall more exact placed boxes for Def 2 than for Def 1.

Since these differences in the results can barely be explained by “technical” reasons (the basic structure of the two final models is, as already mentioned, the same), the differences can most likely be explained with some kind of human bias which is introduced into the data during the labeling process. As mentioned in Section 2, the definitions are only guidelines for a human expert to separate the bands from the rest of the microstructure and not a strict and universally valid criterion. Therefore, it seems reasonable that all or some of the rules of Def 1 tend to introduce more inconsistencies in the respective dataset than Def 2. An additional hint toward these inconsistencies is the more pronounced overfitting for Def 1, observable in Figure 4 and 5. Although it is not clear which part of Def 1 is introducing more of the inconsistency into the dataset, some of it can be traced back to the “continuity condition”, the rule that the bands should range over the whole image. These conditions, although being the most objective criterion in both definitions, can probably be problematic, as shown in Figure 8b,d: In both of images, a false positive is generated by labeling a structure with banded shapes which does not range over the whole image and was therefore not labeled according to Def 1. It is evident that the exclusion of such structures is not comprehensible or not learnable for the model. Additional inconsistencies can be suspected from the box dimensions of the predictions in Figure 8c,d. In these images, the model can find the general position of the band(s) but not the correct dimensions or allocations of the bands.

In contrast to this, the final model for Def 2 does not tend to show these kinds of inconsistencies. One reason can be that Def 2 is stricter than Def 1, which is also underlined by the fact that there are nearly twice as much bands for the first definition (97 vs. 51). With this, it seems that a rather objective criterion like the “continuity condition” is not necessary an indicator for more strict labeling. In general, it can be concluded that the “blockiness” from the latter definition is easier to learn criterion than the “continuity condition”.

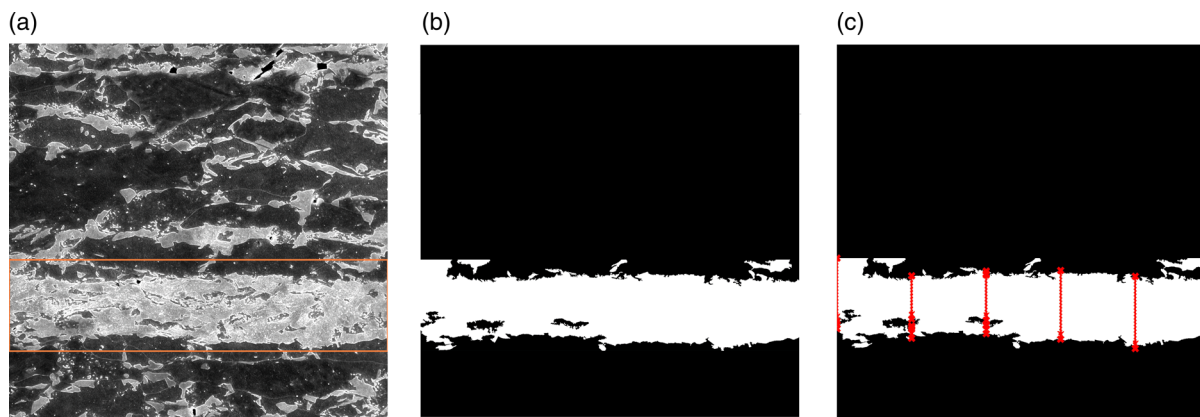
## 6. Generating Data for Representative Volume Elements

(Note: For more detailed information about the DRAGen-RVE Generator, which is used for the creation of the shown RVE, the reader is referred to ref. [18]. The general procedure of the band generation is already described in Pütz et al.,<sup>[22]</sup> and we will briefly review this and explain the link to the Faster R-CNN predictions).

To display the detected bands in DRAGen-RVE, several additional processing steps have to be undertaken, which are schematically displayed in **Figure 9**. The procedure is as follows: In the first step, the trained model is used to predict the martensite bands in an arbitrary number of images. In the second step, the band(s) is/are cropped out of the images based on the bounding box predictions. A combined treatment of despeckling and small hole filling is applied to these images to facilitate statistical characterization of the bands. Thresholds of 1024 and 2048px are used for this. These values are motivated by the fact that the later RVE has much lower resolution than the real image (a common

**Table 4.** Evaluation metrics for the final model for Def 2. The metrics are calculated from the final model w.r.t. to the test set. All values in percent.

Threshold	Precision	Recall	Accuracy
0.5	100.0	95.0	97.5
0.75	90.0	90.0	90.0



**Figure 9.** Schematic overview of the generation of statistical information about the martensite bands from the Faster R-CNN predictions. For more detailed information, see Pütz et al.<sup>[22]</sup> a) Step 1: Predict the martensite band(s) in the SEM images, record the bounding box coordinates further. b) Step 2: Crop out the band, binarize the image, despeckle to remove small objects, and fill the small holes in the image. c) Step 3: Lay straight lines through the band and save the length between the intersection as band thickness (orange lines).

resolution for a DRAGen-RVE is 2), and microstructural components (holes, small islands) with sizes around about 1  $\mu\text{m}$  are not represented by the low resolution of the RVE. After this steps, an image is obtained which contains only the martensite band. To get a statistical representation of the bands, a method is used which is adapted from the ASTM E1268-19, which deals with the “bandiness” of microstructural morphology.<sup>[24]</sup>

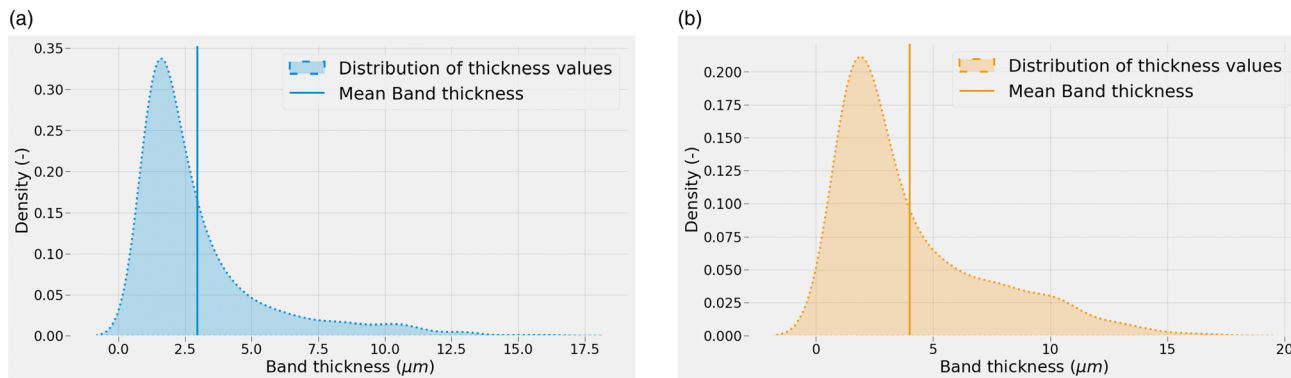
In this method, lines are drawn in y-direction (ND, sheet thickness) of the images, and the intercepts are counted and used for the thickness analysis of the bands. Figure 9c) shows the procedure. The result of this analysis is a table containing the band thickness values based on the Faster R-CNN model, which can then be used for the creation of RVE.

The different definitions will lead to different statistical thickness distributions (see Section 2). The different distributions, using a kernel density estimation (KDE) for displaying, can be found in **Figure 10**. It can be seen that Def 1 skews the calculated band thicknesses to lower values than Def 2. The mean thickness for the bands calculated according to Def 1 is 2.96  $\mu\text{m}$  while the calculated mean thickness for Def 2 is 4.0  $\mu\text{m}$ . Furthermore, Def 1 predicts on average more bands per 50  $\mu\text{m}$  edge length,

namely, 1.4 bands compared to 1.12 bands for Def 2. Another difference is the percentage of martensite inside the bounding boxes which contain the bands (for the whole microstructure the percentage of martensite is around 32%). This value is 44% for Def 1% and 54% for Def 2. (These values are calculated before the despeckling and hole filling operations.) In general, these values indicate that Def 2 is the more strict definition which labels bands that are more different from the rest of the microstructure than Def 1. This was already discussed in Section 5 in relation to the results of the object detection.

## 7. Conclusions

In this article, a machine learning approach was developed to separate martensite bands from martensite islands in a DP800 microstructure. For this, we fine tuned a Faster R-CNN object detector on SEM-Inlens images using two different criteria to define what a martensite band is. The main results can be summarized as follows. 1) A cross-validation shows that there are not much problems with the scarcity of the data. Data augmentation



**Figure 10.** KDEs for the calculated band thickness values for Def 1(left) and Def 2 (right). The vertical line shows the respective mean value. a) Distribution of calculated band thickness values using KDE for Def 1. The mean value (vertical line) is 2.96  $\mu\text{m}$  (b) Distribution of calculated band thickness values using KDE for Def 2. The mean value (vertical line) is 4.0  $\mu\text{m}$ .

techniques help to tackle these problems and to contain tendencies of overfitting. 2) The results are highly dependent on the definition, which should be chosen with great care. The definition is best chosen on the goal of the investigation, for example, if the bands should be displayed in an RVE of a specific size, the required length of the bands should be chosen accordingly. In addition, there is a difference of about 35% on average in the statistically extracted band thickness. 3) The human bias is the biggest problem in this kind of task and depends also on the chosen definition.

In general, the results show that the proposed approach is capable of separating the martensite bands from the remaining microstructure due to arbitrary definitions in a good manner. Also, the approach presented here allows the generation of statistical input data for RVE generation. In doing so, it enables a separate consideration of bands and the rest of the microstructure and the resulting influences. Currently available approaches only allow the general representation of microstructures with different degrees of bandiness. With our method, bands of different definitions can be classified and described and their influence on mechanical properties can be investigated.

Based on this, further improvements can be made to strengthen the results of this study: One the one hand, expert labeling of different experts for the respective microstructure can be tested. The goal of this is to try to exclude “controversial” structures out of the dataset and thereby to contain the human bias which arises if one controversial structure is labeled as a band and another similar structure is not. However, expert labeling should also be accompanied by some kinds of definitions to avoid extremely inconsistent data if each expert is labeling after his or her own definition.

Second, an attempt can be made to generalize the model, for example, by having a look at different DP steel microstructures. With this, maybe more clear criteria can be deduced from the underlying data.

Finally, we only concentrated on 2D SEM images in this study, for the sake of simplicity. In doing so, we ignored the nature of the different bands in the third dimension (which is the transversal direction of the sheet). If the study is extended to 3D microstructural data, additional information can be deduced to improve the consistency of the labeling. However, obtaining large amounts of 3D microstructure data is very costly.

## Acknowledgements

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Projektnummer 278868966, TRR 188, Project B05. The support of this work by the Institute for Physical Metallurgy and Materials Physics (IMM) of RWTH Aachen and Carl Kusche and Maximilian Wollenweber is gratefully acknowledged, who provided the used SEM pictures. The training of the models were performed with computing resources granted by RWTH Aachen University under project thes1181. The authors want to thank the reviewers for their helpful insights which improved the structure of the article.

Open Access funding enabled and organized by Projekt DEAL.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

The data can be found at Niklas Fehleemann's github-page: <https://github.com/nikfeh04?tab=repositories> (Alongside with other data).

## Keywords

dual-phase steels, Faster region-based-convolutional neural networks, machine learning, martensite banding, representative volume elements, transfer learning

Received: November 1, 2022

Revised: January 15, 2023

Published online: March 3, 2023

- [1] R. Liu, A. Kumar, Z. Chen, A. Agrawal, V. Sundararaghavan, A. Choudhary, *Sci. Rep.* **2015**, 5, 11551.
- [2] D. Reimann, K. Nidadavolu, H. ul Hassan, N. Vajragupta, T. Glasmachers, P. Junker, A. Hartmaier, *Front. Mater.* **2019**, 6, 181.
- [3] D. Steinberger, H. Song, S. Sandfeld, *Front. Mater.* **2019**, 6, 141.
- [4] D. Yan, C. C. Tasan, D. Raabe, *Acta Mater.* **2015**, 96, 399.
- [5] K. Gillner, M. Henrich, S. Münstermann, *Int. J. Fatigue* **2018**, 111, 70.
- [6] F. Pütz, M. Henrich, N. Fehleemann, A. Roth, S. Münstermann, *Materials* **2020**, 13, 19.
- [7] C. C. Tasan, M. Diehl, D. Yan, C. Zambaldi, P. Shanthraj, F. Roters, D. Raabe, *Acta Mater.* **2014**, 81, 386.
- [8] S. Heibel, T. Dettinger, W. Nester, T. Clausmeyer, A. E. Tekkaya, *Materials* **2018**, 11, 5.
- [9] J. P. M. Hoefnagels, B. G. Vossen, C. C. Tasan, in *Application of Imaging Techniques to Mechanics of Materials and Structures, Volume 4, Conf. Proceedings of the Society for Experimental Mechanics Series* (Eds: T. Proulx), Springer New York, New York, NY, ISBN 978-1-4419-9528-5 **2013**, pp. 71–77.
- [10] S. Bargmann, B. Klusemann, J. Markmann, J. E. Schnabel, K. Schneider, C. Soyarslan, J. Wilmers, *Prog. Mater. Sci.* **2018**, 96, 322.
- [11] I. M. Gitman, H. Askes, L. J. Sluys, *Eng. Fract. Mech.* **2007**, 74, 2518.
- [12] C. C. Tasan, J. Hoefnagels, M. Diehl, D. Yan, F. Roters, D. Raabe, *Int. J. Plast.* **2014**, 63, 198.
- [13] W. Liu, J. Lian, N. Aravas, S. Münstermann, *Int. J. Plast.* **2020**, 126, 102614.
- [14] N. Vajragupta, P. Wechsuanmanee, J. Lian, M. Sharaf, S. Münstermann, A. Ma, A. Hartmaier, W. Bleck, *Comput. Mater. Sci.* **2014**, 94, 198.
- [15] M. A. Groeber, M. A. Jackson, *Integr. Mater. Manuf. Innovation* **2014**, 3, 56.
- [16] R. Quey, P. R. Dawson, F. Barbe, *Comput. Methods Appl. Mech. Eng.* **2011**, 200, 1729.
- [17] M. Prasad, N. Vajragupta, A. Hartmaier, *J. Open Source Software* **2019**, 4, 1732.
- [18] M. Henrich, F. Pütz, S. Münstermann, *Materials* **2020**, 13, 8.
- [19] M. Fátima Vaz, M. A. Fortes, *Scr. Metall.* **1988**, 22, 35.
- [20] F. Pütz, F. Shen, M. Könemann, S. Münstermann, *Int. J. Fract.* **2020**, 226, 1.
- [21] N. Fehleemann, Y. Sparrer, F. Pütz, M. Könemann, S. Münstermann, *IOP Conf. Ser.: Mater. Sci. Eng.* **2021**, 1157, 012056.
- [22] F. Pütz, N. Fehleemann, V. Göksu, M. Henrich, M. Könemann, S. Münstermann, *Comput. Mater. Sci.* **2023**, 218, 111903.
- [23] C. C. Tasan, J. Hoefnagels, M. Geers, *Scr. Mater.* **2010**, 62, 835.
- [24] E04 Committee, Practice for assessing the degree of banding or orientation of microstructures.
- [25] B. Krebs, L. Germain, A. Hazotte, M. Gouné, *J. Mater. Sci.* **2011**, 46, 7026.

- [26] C. Kusche, T. Reclik, M. Freund, T. Al-Samman, U. Kerzel, S. Korte-Kerzel, *PLoS One* **2019**, 14, 0216493.
- [27] S. Medghalchi, C. F. Kusche, E. Karimi, U. Kerzel, S. Korte-Kerzel, *JOM* **2020**, 72, 4420.
- [28] P. Trampert, D. Rubinstein, F. Boughorbel, C. Schlinkmann, M. Luschkova, P. Slusallek, T. Dahmen, S. Sandfeld, *Crystals* **2021**, 11, 258.
- [29] T. Strohmman, K. Bugelnig, E. Breitbarth, F. Wilde, T. Steffens, H. Germann, G. Requena, *Sci. Rep.* **2019**, 9, 19611.
- [30] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, *Sci. Rep.* **2018**, 8, 2128.
- [31] C. F. Kusche, F. Pütz, S. Münstermann, T. Al-Samman, S. Korte-Kerzel, *Mater. Sci. Eng., A* **2021**, 799, 140332.
- [32] MATLAB, 9.9.0.1467703 (R2020b), The MathWorks Inc., Natick, MA **2020**.
- [33] Deep Learning, *Adaptive Computation and Machine Learning Series*, MIT Press Ltd, Cambridge, MA **2017**.
- [34] O. Ronneberger, P. Fischer, T. Brox (Preprint) arXiv:1505.04597, v1, submitted: May **2015**.
- [35] Z. Zou, Z. Shi, Y. Guo, J. Ye (Preprint) arXiv:1905.05055, v2, submitted: May **2019**.
- [36] J. Redmon, S. Divvala, R. Girshick, A. Farhadi (Preprint) arXiv:1506.02640, v5, submitted: May **2016**.
- [37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár (Preprint) arXiv:1708.02002, v2, submitted: Feb. **2018**.
- [38] R. Girshick, J. Donahue, T. Darrell, J. Malik, (Preprint) arXiv:1311.2524, v5, submitted: Oct. **2014**.
- [39] R. Girshick, (Preprint) arXiv:1504.08083, v2, submitted: Sept. **2015**.
- [40] S. Ren, K. He, R. Girshick, J. Sun (Preprint) arXiv:1506.01497, v3, submitted: Jan. **2016**.
- [41] T. Yin, J. Yang, in *2021 7th Inter. Conf. on Computing and Artificial Intelligence*. ACM, New York, NY, ISBN 9781450389501, **2021**, pp. 15–20.
- [42] S.-J. Oh, M.-J. Jung, C. Lim, S.-C. Shin, *Appl. Sci.* **2020**, 10, 8629.
- [43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimselshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, *Advances in neural information processing systems*, 32 **2019**, <https://doi.org/10.48550/arXiv.1912.01703>.
- [44] K. He, X. Zhang, S. Ren, J. Sun (Preprint) arXiv:1512.03385, v1, submitted: Dec. **2015**.
- [45] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, *Information* **2020**, 11, 125.