Advanced Steel Microstructural Classification by Deep Learning Methods

Seyed Majid Azimi[1,2,4,*], Dominik Britz[2,3], Michael Engstler[2,3], Mario Fritz[1], and Frank Mücklich[2,3]

[1]Max Planck Institute for Informatics, Computer Vision and Multimodal Computing, Saarbrücken, Germany
[2]Material Engineering Center Saarland, Saarbrücken, Germany
[3]Saarland University, Chair of Functional Materials, Saarbrücken, Germany
[4]German Aerospace Center (DLR), Remote Sensing Technology Institute, Weßling, Germany (current affiliation)
[*]seyedmajid.azimi@dlr.de

ABSTRACT

The inner structure of a material is called microstructure. It stores the genesis of a material and determines all its physical and chemical properties. While microstructural characterization is widely spread and well known, the microstructural classification is mostly done manually by human experts, which gives rise to uncertainties due to subjectivity. Since the microstructure could be a combination of different phases or constituents with complex substructures its automatic classification is very challenging and only a few prior studies exist. Prior works focused on designed and engineered features by experts and classified microstructures separately from the feature extraction step. Recently, Deep Learning methods have shown strong performance in vision applications by learning the features from data together with the classification step. In this work, we propose a Deep Learning method for microstructural classification in the examples of certain microstructural constituents of low carbon steel. This novel method employs pixel-wise segmentation via Fully Convolutional Neural Networks (FCNN) accompanied by a max-voting scheme. Our system achieves 93.94% classification accuracy, drastically outperforming the state-of-the-art method of 48.89% accuracy. Beyond the strong performance of our method, this line of research offers a more robust and first of all objective way for the difficult task of steel quality appreciation.

1 Introduction

Steel is still one of the most important and extensively used classes of materials because of its excellent mechanical properties while keeping costs low which gives a huge variety of applications[1, 2]. The mechanical properties of steel are mainly determined by its microstructure[3] shown in Figure 1, so that the performance of the material highly depends on the distribution, shape and size of phases in the microstructure[4]. Thus, correct classification of these microstructures is crucial[5]. The microstructure of steels has different appearances, influenced by a vast number of parameters such as alloying elements, rolling setup, cooling rates, heat treatment and further post-treatments[6]. Depending on how the steel is produced due to these parameters, the microstructure consists of different constituents such as ferrite, cementite, austenite, pearlite, bainite and martensite[7] shown in Figure 1.

This motivation leads us to use Deep Learning methods which are recently grabbing the attention of scientists due to their strong ability to learn high-level features from raw input data. Recently, these methods have been applied very successfully to computer vision problems[8, 9]. They are based on artificial neural networks such as Convolutional Neural Networks (CNNs)[9]. They can be trained for recognition and semantic pixel-wise segmentation tasks. Unlike traditional methods in which feature extraction and classification are learnt separately, in Deep Learning methods, these parts are learnt jointly. The trained models have shown successful mappings from raw unprocessed input to semantic meaningful output. As an example, Masci et al.[10] used CNNs to find defects in steel. In this work, we show that Deep Learning can be successfully applied to identify microstructural patterns. Our method uses a segmentation-based approach based on Fully Convolutional Neural Networks (FCNNs) which is an extension of CNNs accompanied by a max-voting scheme to classify microstructures. Our experimental results show that the proposed method considerably increases the classification accuracy compared to state of the art. It also shows the effectiveness of pixel-based approaches compared to object-based ones in microstructural classification.

2 Related Works

Based on the instrument used for imaging, we can categorize the related works into Light Optical Microscopy (LOM) and Scanning Electron Microscopy (SEM) imaging. High-resolution SEM imaging is very expensive compared with LOM imaging

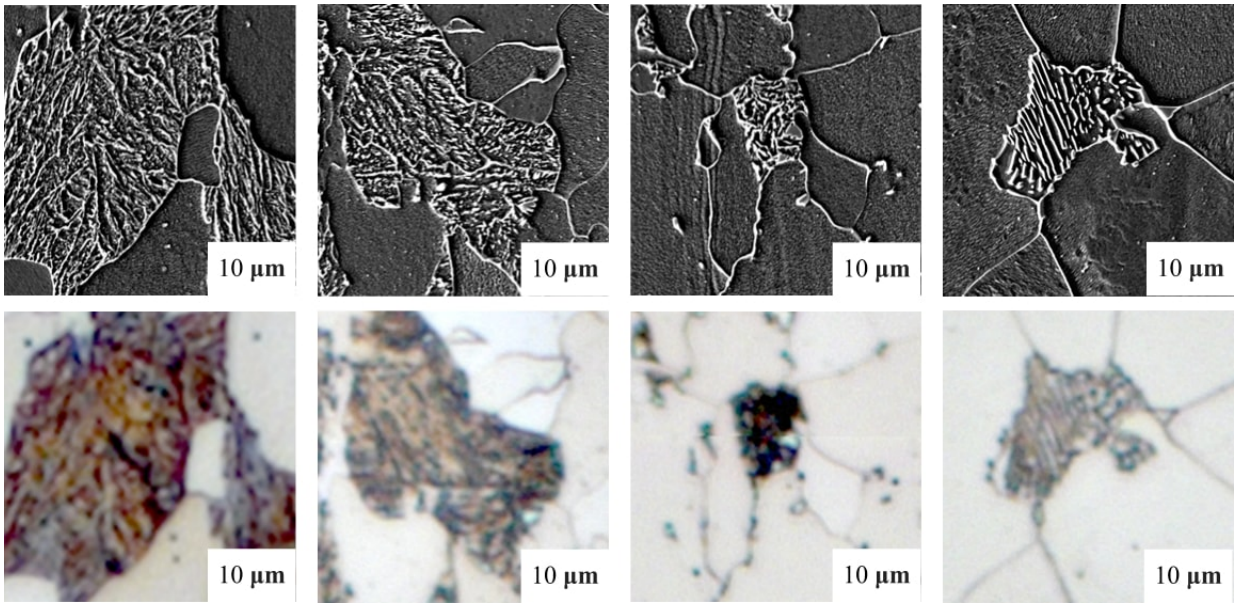


Figure 1. *Some examples of different microstructure classes. In columns from left to right: martensite, tempered martensite, bainite and pearlite phases or rather constituents as "objects" (second-phase) have been illustrated. Ferrite is the matrix phase in these images, having the role of the background. The upper row contains images taken by Scanning Electron Microscopy (SEM) and lower row taken by Light Optical Microscopy (LOM).*

in terms of time and operating costs. However, low-resolution LOM imaging makes distinguishing microstructures based on their substructures even more difficult. Nowadays, the task of microstructural classification is performed by observing a sample image by an expert and assigning one of the microstructure classes to it. As experts are different in their level of expertise, one can assume that sometimes there are different opinions from different experts. However, thanks to highly professional human experts, this task has been accomplished so far with low error which is appreciated. Regarding automatic microstructural classification, microstructures are typically defined by the means of standard procedures in metallography. Vander Voort [11] used Light Optical Microscopy (LOM) microscopy, but without any sort of learning the microstructural features which is actually still the state of the art in material science for classification of microstructres in most institutes as well as in industry. His method defined only procedures with which one expert can decide on the class of the microstructure. Moreover, additional chemical etching[12] made it possible to distinguish second phases using different contrasts, however etching is constrained to empirical methods and can not be used in distinguishing various phases in steel with more than two phases. Nowadays, different techniques and approaches made morphological or crystallographic properties accessible[4, 13, 14, 15, 16, 17, 18]. Any approach for identification of phases in multiphase steel relies on these methods and aims at the development of advanced metallographic methods for morphological analysis purposes using the common characterization techniques and were accompanied with pixel- and context-based image analysis steps.

Previously, Velichko et al.[19] proposed a method using data mining methods by extracting morphological features and a feature classification step on cast iron using Support Vector Machines (SVMs) - a well established method in the field of machine learning [20]. More recently, Pauly et al.[21] followed this same approach by applying on a contrasted and etched dataset of steel, acquired by SEM and LOM imaging which was also used in this work. However, it could only reach 48.89% accuracy in microstructural classification on the given dataset for four different classes due to high complexity of substructures and not discriminative enough features.

Deep Learning methods have been applied in object classification and image semantic segmentation for different applications. AlexNet, a CNN proposed by Alex Krizhevsky et al.,[22] with 7 layers was the winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[23]in 2012 which is one of the most well-known object classification task challenges in computer vision community. It is the main reason that Deep Learning methods drew a lot of attention. AlexNet improved the accuracy ILSVRC2012 by 10 percent points which was a huge increase in this challenge. VGGNet, a CNN architecture proposed by Simonyan et al.[8] has even more layers than AlexNet achieving better accuracy performance. Fully Convolutional Neural Networks (FCNNs) architectures, proposed by Long et al.,[24] is one of the first and well-known works to adapt object classification CNNs to semantic segmentation tasks. FCNNs and their extensions to the approach are currently the state-of-the-art in semantic segmentation on a range of benchmarks including Pascal VOC image segmentation challenge [25]

or Cityscape [26].
Our method transfers the success of Deep Learning for segmentation tasks to the challenging problem of microstructural classification in the context of steel quality appraisal. It is the first demonstration of a Deep Learning technique in this context that in particular shows substantial gains over the previous state of the art.

3 Review of recent Deep Learning techniques in computer vision

Previous work in the context of steel microstructural classification relies on hand-designed features. Engineering high-level features e.g., morphological [21] features require complex feature extraction algorithms and a tedious trial-and-error process of finding good features in the first place. Deep Learning methods are capable of learning complex features from raw input data that turn out to also be superior across a wide range of application domains. These methods are inspired by neural networks and an "end-to-end" learning paradigm. Unlike classical methods, feature extraction and classification is done simultaneously in Deep Learning methods and optimized jointly. This is facilitated by composing complex, parameterized functions from simple, efficient, piece-wise differentiable building blocks. Therefore, training the whole system is facilitated by gradient descent on the parameters using a recursive gradient computation via the chain rule of differentiation (this is called back propagation algorithm [27]). Among these Deep Learning methods, Convolutional Neural Networks (CNNs) and their modification, Fully Convolutional Neural Networks (FCNNs) have been shown particularly successful for image classification and segmentation. CNNs were developed originally for vision tasks. The basic concept of CNNs dates back to 1980[9, 28], but due to the increase in computational resources, available data and better learning algorithms, they have recently attained a new level of quality across a range of domains. Therefore, CNNs are the state-of-the-art approaches in many vision applications. As images are of high dimensionality applying traditional neural networks with fully connected neurons to process visual data would lead to a huge number of trainable parameters. Furthermore, they are not able to exploit the natural structures in the images like correlation among neighboring pixels and stationary image statistics. Typical Convolutional Neural Networks consist of multiple, repeating components that are stacked in layers: convolution, pooling, fully connected and classifier layers.
Convolution Layer convolves the input data with a linear convolution filter as shown in Equation 1:

$$(h_k)_{ij} = (W_k * x)_{ij} + b_k \tag{1}$$

where $k = 1, \ldots, K$ is the index of the k-th feature map in convolution layer and (i, j) is the index of neurons in the k-th feature map and x represents the input data. W_k and b_k are trainable parameters (weights) of linear filters (kernel) and bias for neurons in the k-th feature map respectively. $(h_k)_{ij}$ is the value of the output for the neuron in the k-th feature map with position of (i, j). The spatial 2D convolution operation between the input data and the feature map has been represented by "$*$".
Pooling Layer is a nonlinear down-sampling layer which either takes maximum or average values in each sub-region of the input data. Today's CNNs typically employ a maximum pooling called "max-pooling" layer in order to achieve invariance to small shifts in the feature maps.
Fully-connected Layer is a classic neural network layer where the features of the next layer are a linear combination of the features of the previous layer as shown in Equation 2:

$$y_k = \sum_l W_{kl} x_l + b_k \tag{2}$$

where y_k represents the k-th output neuron and W_{kl} is the kl-th weight between x_l and y_k.
Activation Function usually follows a pooling or fully connected layer and introduces a nonlinear activation operation like a sigmoid or rectified linear unit (ReLU). The ReLU function $relu(x) = \max(0, x)$ are most common as the gradient is piece-wise constant and does not vanish for high activations (in contrast to e.g., sigmoid).
Classifier Layer is the last layer of the network and computes a class posterior probability of the input image. The most widely used classifier layer in CNNs is the softmax function. The vector of real values between $(0, 1)$ generated by this function denotes a categorical probability distribution shown by Equation 3 for the j-th class and an input vector X.

$$P(y = j|X; W, b) = \frac{\exp^{X^T W_j}}{\sum_{k=1}^{K} \exp^{X^T W_j}} \tag{3}$$

Loss Layer is used to measure the difference between true class labels and corresponding predicted class labels. The most common loss layer for classification is the cross-entropy loss. The cross-entropy loss is shown in Equation 4:

$$\text{Cross-entropy Loss Function} = -\sum_x P^{'}(x) \log P(x) \tag{4}$$

in which the softmax classifier ($P(x)$) is minimizing the cross-entropy between the "true" one-hot encoded distribution of data($P^{'}(x)$) and the predicted class probabilities.

CNN Training is done end-to-end which means without separation between the feature extraction and the classification step. They receive raw input data e.g., image pixels, to produce semantic outputs. CNNs are trained to learn the hidden pattern in the data by using a training set. A loss function $\mathscr{L}$ measures how far the outputs of the network from the correct output is. This optimization problem is solved by gradient descent[9] and a well-known process called Back-Propagation[9] to propagate the loss function gradient to previous layers. The gradient descent is performed based on the loss function gradient $\frac{\delta \mathscr{L}}{\delta w_{ij}}$. Then, the weights are adapted in order to decrease the loss function output by modifying into the direction opposite to the direction of increasing gradient with a step size (learning rate) represented by η in Equation 5. Learning rate does not have a unit. It is a user defined parameter. It determines how much gradient of loss function is applied to update weights in each back-propagation step during CNN training phase.

$$w_{ij}^{\textbf{new}} = w_{ij}^{\textbf{old}} - \eta \frac{\delta \mathscr{L}}{\delta w_{ij}} \tag{5}$$

Dropout Layer[29] is a technique to improve the generalization power of CNNs by randomly ignoring (dropping) neurons and their corresponding parameters from the network architecture only during the training phase.

As annotated SEM images are rare, most likely a training network on such dataset will lead to overfitting to noise present in the training set. To address this problem, networks which have already been trained using large training datasets like ImageNet are trained on the new dataset. This trick is known as "Transfer Learning" or "fine tuning". Using this technique, we can initialize the weights more efficiently. Therefore, it can be assumed that the network is already close to the best local minimal solution and needs far less training data to converge. From pre-trained CNNs, features can also be extracted without fine tuning. In this case, the network is not fine tuned. Instead, output of fully connected layers before the classification layer is considered as feature vector. These features known as DeCAF[30] and can be classified with e.g., SVMs. Another trick in case of utilizing small datasets is to artificially enlarge the training set e.g., by flipping or rotating while preserving the class labels. This trick is known as "Data Augmentation".

Network Architectures: In the following, we describe the three specific CNN architectures (with increasing depth) that we use for classification: *CIFARNet* is a CNN with three convolutional and max-pooling layers with two fully-connected layers. It is a modified version of LetNet[9] proposed by Lecun et al., containing 431K parameters to which the ReLU activation layer and dropout layer (will be described in the following) have been added.

AlexNet, proposed by Alex Krizhevsky et al.,[22] is a deep CNN with 60 million parameters.

AlexNet is deeper than CIFARNet. It has eight layers including five convolutional, three max-pooling and three fully-connected layers. *VGGnet* was proposed by Simonyan et al.[8] and it is even deeper with 13 convolutional, five pooling and three fully-connected layers known as VGG16 with 138 million parameters. Another version called VGG19 has 19 layers. VGG19 network was able to achieve empirically better performance than CIFARNet and AlexNet. In VGGnet, a 3x3 convolution kernel size was applied which resulted in fewer parameters, but with the same support.

3.1 Fully Convolutional Neural Networks (FCNNs)

While CNNs have been shown successful for image classification, we are also interested in predicting a set of semantic classes for each pixel which is called semantic segmentation. CNNs can be extended to perform this task by removing the fully-connected layers and "sliding" the CNN over a larger image. However, the resulting semantic segmentation would not quite be of the same resolution as the original image. Therefore, Long et al.[24] proposed Fully Convolutional Neural Networks(FCNNs) with an additional up-sampling layer.

Up-sampling layer can be achieved by simple bilinear interpolation (which is differentiable). Today's FCNNs learn also the up-sampling kernel which is parameterized as shown in Equation 6:

$$y_{ij} = \sum_{\alpha,\beta=0}^{1} |1-\alpha-\{i/f\}| \ |1-\beta-\{i/j\}| \ x_{\lfloor i/f \rfloor+\alpha,\lfloor j/f \rfloor+\beta}, \tag{6}$$

where y and x are the input and output of the up-sampling layer, and f and α,β stand for the up-sampling factor and pixel fraction parts, respectively. One can consider an up-sampling layer with factor f as a convolution operation, but instead of an integer stride it has a fractional input stride of $1/f$.

To capture different information about the input data by preserving the dimensions of input feature maps, we can fuse different pooling layers together known as "skip layers". FCNNs can be seen as an encoder-decoder system in which convolutional layers do encoding of input images by extracting high-level learned features and deconvolution layers do decoding on these

features to present semantic segmentation of the input. As image semantic segmentation datasets are typically small, these networks are frequently pre-trained on object recognition datasets and then fine-tuned to perform the segmentation task.

4 Methods

In this section, we describe our methods to classify microstructures in steel. First, we applied Deep Learning methods to classify each cropped steel object as illustrated in Figure 2 from SEM or LOM images which we call object-based microstructural classification. Then we explain our main methods which classify each pixel as one of a microstructure class and then we classify each object by considering the classes of pixels inside the object. To avoid misunderstandings, "substructure" and "texture" is used equally, owed the different current languages in material science and computer vision. All experimental protocols have been approved by Material Engineering Center Saarland (MECS) and Max Planck Institute for Informatics (MPI) institutes. The methods were carried out in accordance with the relevant guidelines and regulations.

4.1 Object-based Classification of Microstructures with CNNs

The first approach we used, was the classification of microstructures (second/dual phases) by CNNs. However, CNNs work on images of fixed size. Therefore, we normalized the images for each structure by cropping objects from the images, resized them to a fixed size and then applied the CNN classifier.

Masking: By applying a user-defined threshold on pixel intensity in LOM image, binary segmented LOM as shown in Figure 2 can be computed. Corresponding SEM image which has already been registered with LOM image from the same area are masked using the mentioned binary mask. In the resulting masked SEM all of matrices (here ferrite) are masked as illustrated in Figure 2. In addition, by using this mask, each constituent can be localized.

Cropping: Using location information obtained from the binary segmented LOM, each constituent (object) in the masked SEM image is cropped and separated from the main image as shown in the part of "cropped object" in Figure 2.Hence, the cropping operation is systematic. These operations have been described with more details in Britz et al.[31]

Warping: Due to fully-connected layers in CNNs, input images should be warped to a fixed size e.g., 224x224px for VGG16 network. The cropped objects should be then warped to a fix size.

Classification: Each warped constituent (object) image is then given to CNN as input to be classified. In this way, the system performance is comparable to the previous work of Pauly et al.[21] As the objects have been automatically warped and split randomly into training and test set, their distribution can be assumed identical independently distributed. We considered three possible techniques of using CNNs for image classification: (I) full-training (from scratch) CNNs, (II) fine-tuning CNNs and (III) DeCAF features with SVM classifier. Since in this case, classification was done directly based on the object input image, we refer to this system as the object-based CNN microstructural classification.

The (I) technique is a network which was trained with random initialized parameters. In this method we are free to choose the size of the input image. The (II) strategy was using *transfer learning*. However, in this case, we were limited to the input size of the CNN.

The first and second strategy is illustrated in Figure 2. The third strategy is similar except that classification is done by SVMs rather than a softmax layer. The output of the object-based CNN before classification is $P(C_i|O)$, where C_i is the class of each phase, O stands for the observation or the input image, and P is the posterior probability that the input image belongs to class i. Classification is performed by choosing the class with the highest probability.

4.2 Segmentation-based Classification of Microstructures with FCNNs

Resizing of each cropped object image to a fixed size, as required in the object-based CNN approach, could destroy valuable information related to the phase texture by heavy distortion. On the other hand, pixel-wise classification (segmentation) can work with any image size. Thus, we propose a SEM or LOM image segmentation-based microstructural classification approach using a FCNN and max-voting scheme to classify each object. The processing pipeline for this approach is illustrated in Figure 3. We refer to this approach as max-voted FCNN (MVFCNN). Using this method, SEM or LOM images are classified pixel-wise. In our experiments, we used a network architecture proposed by Long et.al.[24]. The architecture is almost the same as VGG16, except a converting from fully-connected layers to convolutional ones and up-sampling layers plus skip layers. They used skip layers to fuse coarse and local appearance information to improve the semantics. By using no skip layer, the network is called FCN-32s, with skip layer A and with A and B together denoted in Figure 3 network is called FCN-16s and FCN8s respectively. We used cropped raw SEM or LOM images as input to FCNNs. For pixel-wise microstructural classification, we can not use the original image sample due to the big size of samples(7000x8000 px) and GPU memory limit. Therefore, original image should be cropped into small patches and each patch should be segmented separately. Patch cropping is done using sliding window technique (e.g., Dalal et al.[32]). We consider a window of HxW size illustrated in Figure 3. This window slides over the original image with a horizontal and vertical step sizes. We call these step sizes as stride parameters. In each step, the original image is cropped using the sliding window. Then the cropped patch is given to FCNN as input image. We do

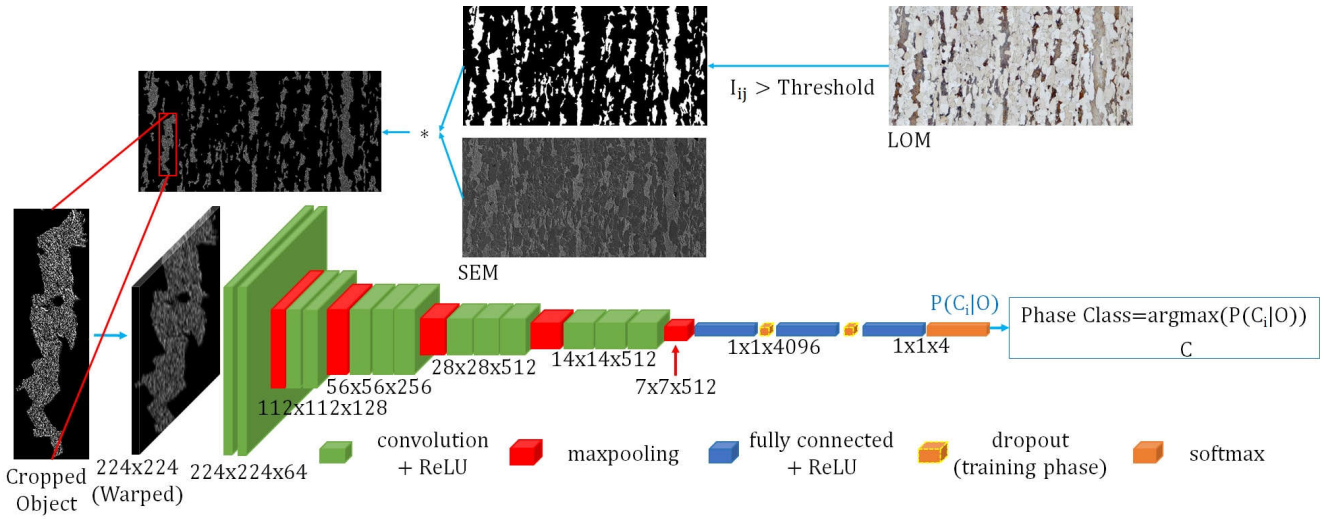


Figure 2. *Workflow of object-based classification approach using CNNs. In this figure one object in the SEM image is cropped and classified using a trained CNN. 224x224px is the fixed input size of the VGG16 network and "C" stands for class.*

this operation for all of patches to cover the whole input image. In other words, the whole original input image will be given as separate input patches to the network. The maximum size of the cropped images is determined by the GPU memory.

The output of FCNN is a 3D matrix with the number of channels equal to the number of classes. Each pixel in this matrix has a value representing the score confidence or posterior probability (output of the softmax classifier function) for the corresponding microstructure class C_i. The pixel-wise classification step is then performed by choosing the class for each pixel with the highest posterior probability. Afterwards, all of the segmented patches belonging to the original input image are stitched together as illustrated in Figure 3. In order to classify objects (microstructures) rather than pixels, a max-voting policy is applied on each object, assigning it to the class of the majority of the pixels. In other words, the microstructure class with the maximum classified pixels inside an object will be assigned to that object. The location information of objects is obtained using a binary LOM image. The motivation for using this aggregation step was that by using stitched patches we can decide the class of each object instead of only a part of it.

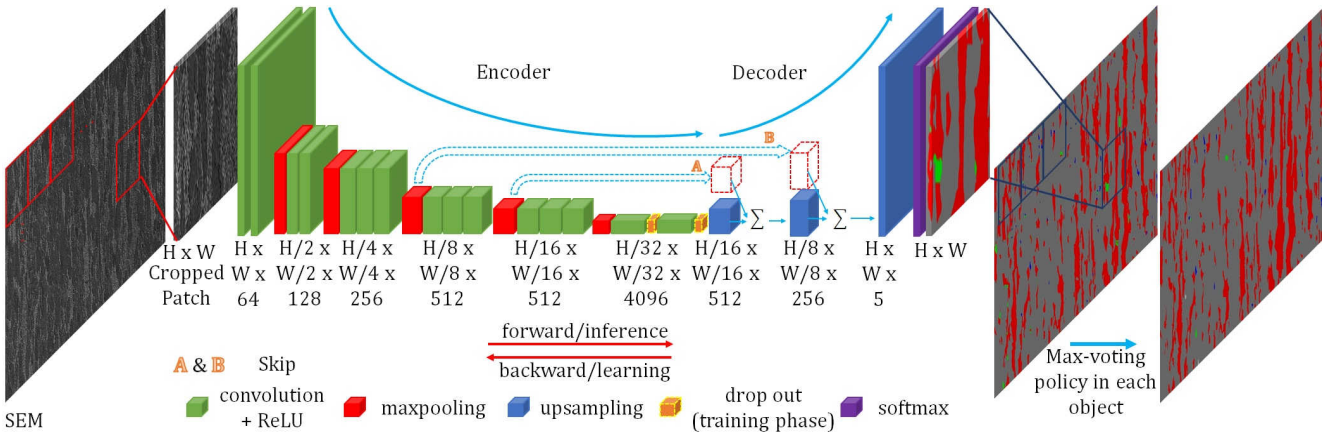


Figure 3. *Workflow of max-voted segmentation-based microstructural classification approach using FCNNs (MVFCNN). In this figure, the input image is an SEM image. The input image is first cropped. Then cropped patches are given to the FCNN part of the MVFCNN algorithm. The segmented crops are stitched together by FCNN. In the last step, the Max-voting policy is applied on the resulting stitched image. The max-voted output is used to classify microstructure objects. H and W represent height and width and the third number is the number of the feature map.*

4.3 Implementation Details

In order to train and test CNNs and FCNNs, Caffe [33] framework and a K40m NVIDIA GPU was used. Caffe framework is a library in which most of the fundamental layers of neural networks have been implemented efficiently by C++ and Cuda programming languages.

4.3.1 Training object-based CNN:

All of the cropped object images were resized to 224x224px which is the fixed input size of VGG16. We also considered this size of input for training networks from scratch. We used a fixed learning rate of 0.001, a momentum of 0.9 and weight decay of 0.004 in stochastic gradient descent algorithm. The training iterations continue until the training loss reached a plateau. For training CIFARNet from scratch, the standard deviation of Gaussian noise for initial random weights for the first convolutional layer is 0.0001 and for the rest is 0.01. For fine tuning, a pre-trained VGG16 network was used. We initialized the last fully-connected layer with random Gaussian noise with standard deviation of 0.01. The learning rate of 0.0001 (chosen on validation set) was used to train CIFARNet and VGG16 respectively.

4.3.2 Using pre-trained extracted features (DeCAFs):

To classify DeCAF features using SVMs, we trained a multi-class SVM with RBF kernel with extracted features from pre-trained VGG19[8] network. In VGG19 architecture, a fully-connected layer before the classification layer (size of 1 x 1 x 4096px) was considered as the feature vector. Therefore, the feature vector is a 1x1x4096 dimension vector.

4.3.3 Training MVFCNN:

In training MVFCNN, we used learning rate of 10^{-10}, 10^{-11} and $3*10^{-12}$ to train FCN-32s, FCN-16s, and FCN-8s, respectively. A bigger learning rate causes the training loss to explode. The momentum of 0.9 with weight decay of $5*10^{-4}$ was considered. Regarding input images, patches were cropped with 1000×1000 px size with a batch size of 1, due to memory issues. We first trained a FCN-32s model, and then added a skip layer (FCN-16s) and fine-tuned it. Afterwards, another skip layer (FCN-8s) was added to fine-tune the final model. Direct training of FCN-8s gave worse results. A pre-trained FCN-32 model was trained with an ImageNet dataset. The network was trained with 7000 iterations for ~ 4.5 days. The inference time for a 1000x1000 px input image was ~ 600 ms.

4.3.4 Class Balancing and Data Augmentation

In order to address the problem of class unbalance in the dataset, in the MVFCNN method cropping was carried out for different classes with different stride (step size) parameters in horizontal and vertical directions which in the end all of classes had the same number of patchesi.e. the class with least number of images had smaller stride than the class with the largest number of training images. Stride parameters are the parameters determining the steps using which patches are cropped. For example if the horizontal stride is 100px, it means when a patch is cropped, the next patch will be cropped with a 100px step. The same goes for the vertical direction. In our experiments, we chose different stride parameters corresponding to the number of images for each class. Stride parameters in horizontal and vertical directions were chosen the same. For an example, an image of 7000x8000 px includes $7*8=56$ patches with 1000x1000 px size stitched together. By using a stride parameter of 100 px, one can produce $61*71=4331$ patches each with the size of 1000x1000 px. Resulting cropped patches were also rotated by 90°, 180° and 270° to augment the dataset. In this case, the number of the training images can increase three times.

5 Results

5.1 Dataset

In our experiments, we use a steel image dataset [21] provided by Material Engineering Center Saarland (MECS) in Saarbrücken, Germany.

The dataset contains registered images of steels taken by LOM and SEM and the corresponding binary LOM images. The steel samples were first polished and etched according to Britz et al.[18] and Pauly et al.[21] In total, 21 LOM and SEM images with an average size of 7000x8000px were available. LOM and SEM images for each microstructure class are shown in Figure 4. For ground truth, the procedure according to Britz et al.[31] was applied. Afterwards a group of material experts and metallographers assign the objects of the second phase to the mentioned phases/constituents according to Aarnts et al[5]. In other words, the assignment of the images to each microstructural class (ground truth) has been made by material science experts and metallographers. For example, if one SEM image contains martensitic and ferritic constituents, a group of material science experts have assigned martensite and ferrite labels to the "objects" or rather grains which the mentioned sample contains. Informed consent has been obtained from all material expert co-authors from MECS.

The microstructures contain a ferritic matrix as first phase and a pearlitic, martensitic or bainitic microstructures as second constituent. Therefore, objects are also referred to as two-phase steel images with two constituents. 21 images are distributed into martensitic (11 samples), tempered martensitic (2 samples), bainitic (4 samples) and pearlitic (4 samples). A further

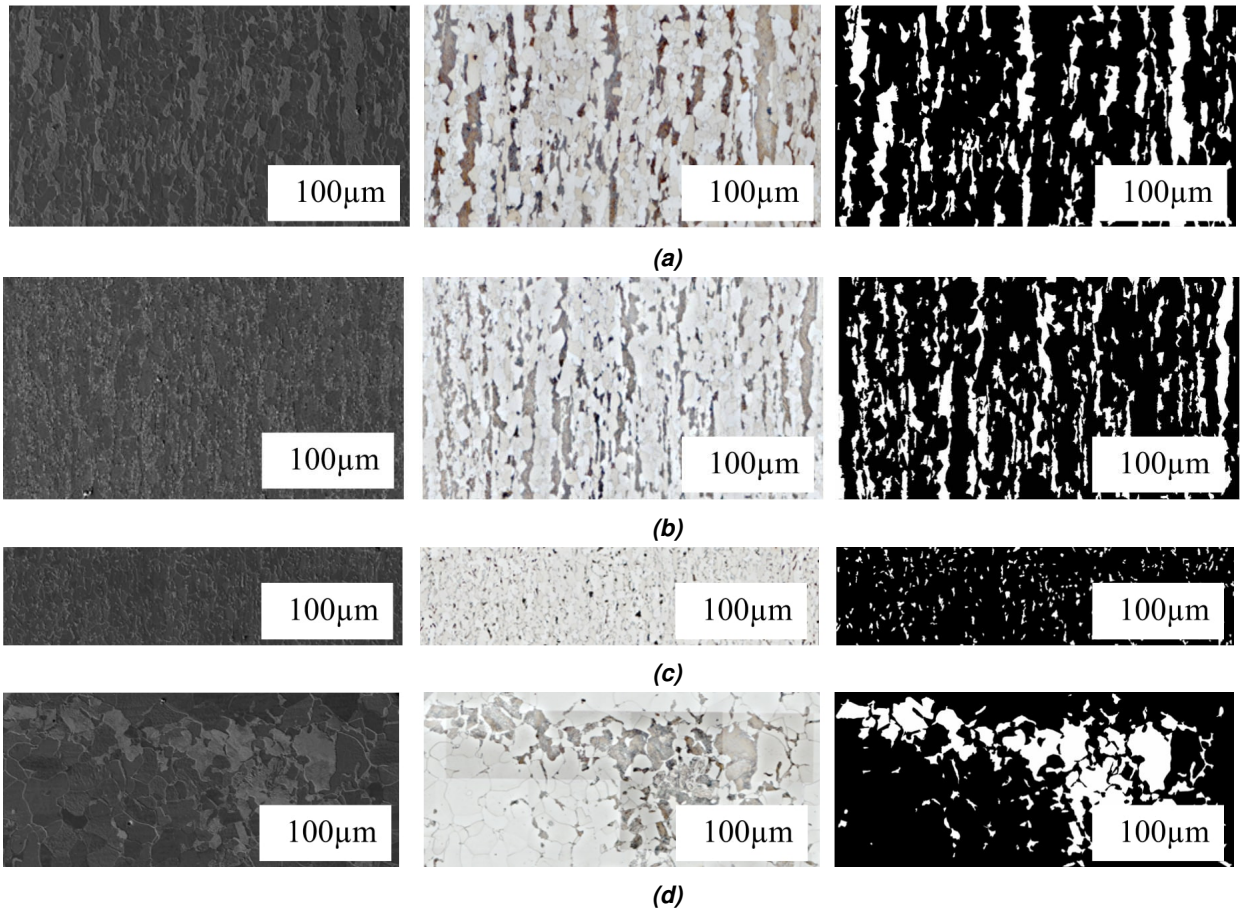


Figure 4. *SEM and LOM example images for each microstructure class with ferrite as matrix and with diameter of up to 100 µm. The columns show SEM, LOM and segmented LOM images from left to right. Size ratios of samples have been preserved. The sub-images are corresponding to (a) martensitic, (b) tempered martensitic, (c) bainitic and (d) pearlitic microstructures.*

subdivision for the bainite (upper, lower, granular etc. according to Zajac et al.[34]) will be considered in the next step after this proof of concept. The dataset split is image-based with 11 training images and 10 test images which results in 2831 training and 2262 test objects. It should be mentioned that in the test set, one of two bainite samples is a bainitic sample which was normalized afterwards and therefore, it shows a different appearance despite being in the same class.

5.2 Task Definition and Metrics

In the microstructural classification problem, the goal is to classify constituents (object) inside steel images based on microstructure classes. These objects are grains within the microstructure which can be considered as "background" (matrix (ferrite)) and "foreground" (second constituent in the present microstructure) according to Figure 5. The substructure of such an object can be seen as texture - which should not be confused with the meaning of the term "texture" in Material Science - which will be classified with the correct label in the present work. Therefore, the more objects that are classified correctly, the more accurate the system is.

In addition to the classification task, we also evaluated the semantic segmentation performance using the metrics in Equation 7, 8, 9, and 10. For more information, the reader is encouraged to read the paper by Long et .al.[24] In the following, n_{ij} is the number of the pixel of class $i-$th predicted as class j, n_{cl} is the number of different classes and $t_i = \sum_j n_{ij}$ is the whole pixel number of class i:

- pixel accuracy:

$$\frac{\sum_i n_{ii}}{\sum_i t_i} \tag{7}$$

- mean accuracy:

$$\frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i} \tag{8}$$

- mean intersection over union (mean IU):

$$\frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{9}$$

- frequency weighted intersection over union (frequency weighted IU):

$$(\sum_k t_k)^{-1} \frac{\sum_i t_i n_{ii}}{t_i + \sum_j .n_{ji} - n_{ii}} \tag{10}$$

In microstructural classification via a pixel-based MVFCNN approach, segmented objects are classified with the class that the majority of those pixels vote for (max-voting). Then the evaluation is carried out by enumerating correct classified microstructure objects. After this step, the results of object-based and pixel-based methods are comparable with each other as well as with the state of the art.

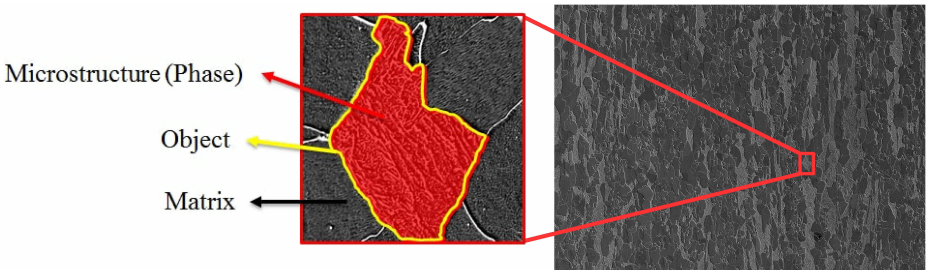


Figure 5. *SEM image of a steel microstructure. The right image is an original SEM image and the left one shows a magnified object. In this example, the substructure (microstructure) is martensitic within a ferritic matrix (background/first phase). The foreground (second/dual phase/constituent) will be considered as "object".*

5.3 Evaluation

In Table 1, comparable results of our experiments with object-based CNN and pixel-based MVFCNN are presented using SEM images. Also the performance of the previous state-of-the-art method of Pauly et al.[21] with an accuracy of 48.89% is shown. In their work, they enumerated the correctly classified microstructural objects by data mining tasks using the same dataset. Hence, the performance of our system both object-based CNN and pixel-based MVFCNN are comparable with theirs. Instead, from-scratch trained CIFAR-Net is able to outperform this method by achieving 57.03% accuracy. Using pre-trained VGG19-Net features with a SVM classifier and a fine tuned VGG16-Net we even achieve a better performance of 64.84% and 66.50%, respectively. All of these methods apply object-based algorithms. However, Table 1 shows that microstructural classification using pixel-based methods can achieve a considerably higher accuracy of 93.94% accuracy which shows that the pixel-based classification of microstructures is a very promising approach.
In Table 2, the effect of data augmentation and fine tuning using SEM images in the MVFCNN approach have been depicted. The results shows that fine tuning and using balanced augmented training data achieves the best results of 93.94%. Among training data and training strategy, using a fine tuned MVFCNN has the most impact of 32.48 percent points improvement compared to MVFCNNs with unbalanced data that are trained from scratch. In Table 3, the accuracy of the methods using LOM images have been provided which are able to achieve up to 70.14% accuracy with a similar configuration as the best results using SEM images. These results indicate that, by considering the available dataset, the LOM is not as useful as the SEM because of the feature size of the objects in this microstructural classification approach. However, there is still a potential to increase the LOM accuracy by taking into account e.g., color etchings like Klemm or contrasting described in the literature [11]. In Table 4, the confusion matrix of the MVFCNN approach (fine tuned with balanced and augmented SEM training data) as the best performing method without matrix (ferrite), is shown. In this matrix, missed objects in segmentation step are not taken into account (#48 objects). That is why the overall accuracy is 95.23% (see Table 4) which is slightly higher than the best classification accuracy of 93.94%, shown in Table 1. The confusion matrix shows the number of samples for

Table 1. *Microstructural classification results using object-based CNN and pixel-based MVFCNN approaches. The results show that object-based classification approaches improve over prior work at most by around 18 percent points. The pixel-based approach has even better performance by around 45 percent points improvement.*

Method	Type	Training Strategy	Accuracy
Pauly et al.[21]	object-based	–	48.89%
CIFAR-Net	object-based	from scratch	57.03%
pre-trained VGG19-Net Features + SVM	object-based	–	64.84%
VGG16-Net	object-based	fine tuning	66.50%
MVFCNN	pixel-based	fine tuning	**93.94%**

each class predicted by the system. Recall and precision numbers show the correct classification percentage of actual classes and predictions, respectively e.g., the network has classified 1190 martensite objects correctly which is 94.97% of the whole martensitic objects, Table 4, column 1. On the other hand, the network has misclassified 24 and 39 martensitic objects as tempered martensite and bainite, respectively. This performance gives 94.97% recall rate and 99.08% precision rate. The dataset is heavily unbalanced, especifically for tempered matersitic samples. Even though there is one training image for tempered martensite, it is a big image for which, after cropping and applying data augmentation techniques, the number of patches are far more than one. In our experiments without data augmentation, we could not get a high precision rate for tempered martensitic samples. However, after having used data augmentation techniques, the precision rate for tempered martensite was improved dramatically. In Table 5, the results of the pixel-wise semantic segmentation with different configurations are presented. In this table, as expected, the FCNN method for pixel-wise segmentation step can achieve the best results. It can achieve a pixel accuracy of 93.92%, a mean pixel accuracy of 76.70%, a mean intersection over union of 67.84% and a frequency weighted intersection over union of 88.81%. In Table 6, the same configurations have been evaluated pixel-wise for accuracy of each class. The matrix has the highest pixel accuracy of 94.22% as expected. And bainite has the lowest pixel accuracy of 37.32%. The matrix is present in all examples which is the reason why the network has learned its structure well - although it has to be mentioned that so far, the grain boundaries are neglected. However, bainite has small objects and also one of two bainite test images is the transformed bainite sample which the network has not seen what leads to a poor performance in this class. Surprisingly, LOM can achieve a pixel accuracy of matrix with 94.11% which is comparable to the SEM images.
In Figure 6a, some successful examples of SEM segmentation using FCNN networks, trained with balanced and augmented training data are shown next to some failure cases in Figure 6b. Regarding the failure case of bainite, it should be noted that the network was trained using isothermally transformed bainitic specimens which showed no failure cases in the test set. The final results of the microstructural classification by stitching and applying a max-voting scheme over the segmented patches are depicted in Figure 7. The final results in this figure show that most of the objects in each microstructure image are classified correctly. If we consider the classification of the overall microstructure of each image, all of the ten test images are classified correctly.

Table 2. *The effect of fine tuning and data augmentation techniques using the MVFCNN approach are depicted. The results show that fine tuning together with data augmentation achieves the best result. However, the effect of data augmentation is not significant.*

Network Architecture	Training Data	Training Strategy	Test Accuracy
MVFCNN	Unbalanced SEM	from scratch	55.50%
MVFCNN	Unbalanced SEM	fine tune MVFCNN	87.98%
MVFCNN	Balanced SEM	fine tune MVFCNN	90.97%
MVFCNN	Balanced and augmented SEM	fine tune MVFCNN	**93.94%**

To compare segmentation performances between segmented LOM and SEM images, in Figure 8a, segmentation of patches of four microstructures using LOM and SEM images by the FCNN approach, trained with balanced and augmented training data have been shown. In most cases, SEM segmentation has a better accuracy, but it is interesting that the network can still learn correct pixel-wise classification from a low-resolution LOM images compared to SEM which suffer from different illumination and stitching artifacts. Figure 8b shows a few noisy sample patches in SEM images with the corresponding segmentations. Noise originates mostly from dirt, dust and sample preparation. The matrix (ferrite) segmentation is considered as background which is also presented showing a very good robustness.

Table 3. *The results of the same experiments with LOM images are provided in this table. The results show inferior performance using LOM instead of SEM images.*

Method	Input Data	Data	Training Strategy	Test Accuracy
CIFARNet	cropped object	LOM	from scratch	51,27%
VGG-19- DeCAF- RBF-kernel SVM	cropped object	LOM	–	56.56%
VGG-16	cropped object	LOM	fine tuning	60,02%
MVFCNN	cropped patch	LOM	fine tuning	70.14%
MVFCNN	cropped patch	SEM	fine tuning	**93.94%**

Table 4. *Confusion Matrix of the best MVFCNN approach. The matrix shows the number of samples for each class predicted by the system. Due to an unbalanced multi-class problem, percentage numbers for each class show normalized recall rates. Note: #48 not-segmented objects have not been considered. Overall accuracy is 93.94%.*

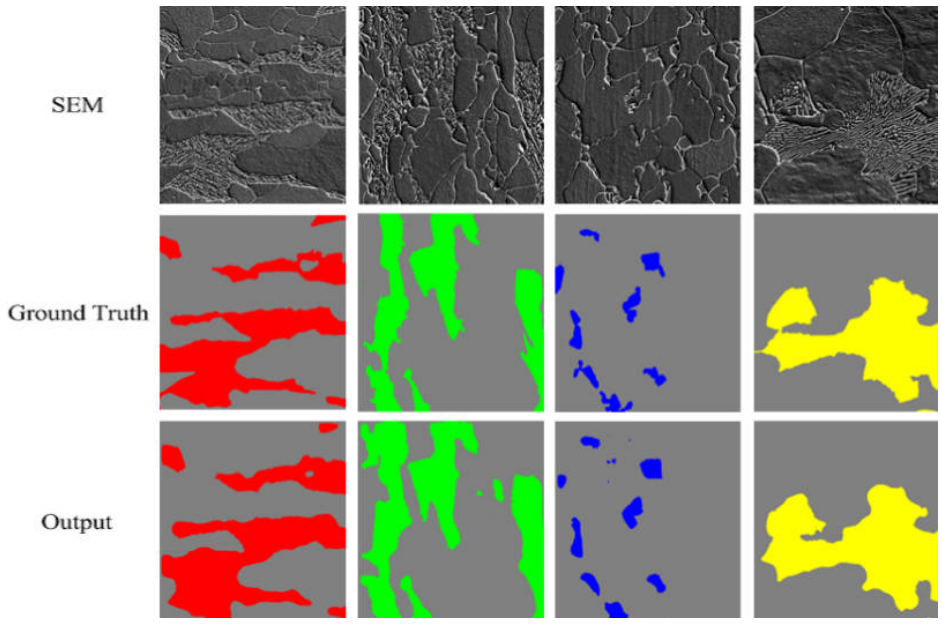
		Actual Class Labels				
		Martensite	Temp. Martensite	Bainite	Pearlite	**Class Precision**
Predicted Class Labels	Martensite	1190 94.97%	0 0.00%	11 3.19%	0 0.00%	99.08%
	Temp. Martensite	24 1.92%	268 97.81%	0 0.00%	0 0.00%	91.78%
	Bainite	39 3.11%	6 2.19%	325 94.20%	16 4.80%	84.19%
	Pearlite	0 0.00%	0 0.00%	9 2.61%	317 95.20%	97.23%
	Class Recall	94.97%	97.81%	94.20%	95.19%	**Total Accuracy** 95.23%

Table 5. *Evaluation of the segmentation-based approach using FCNNs with different training data and training strategies. As the results show, using SEM images, fine tuned networks and augmented data, the best performance can be achieved. However, LOM has an inferior performance compared to SEM.*

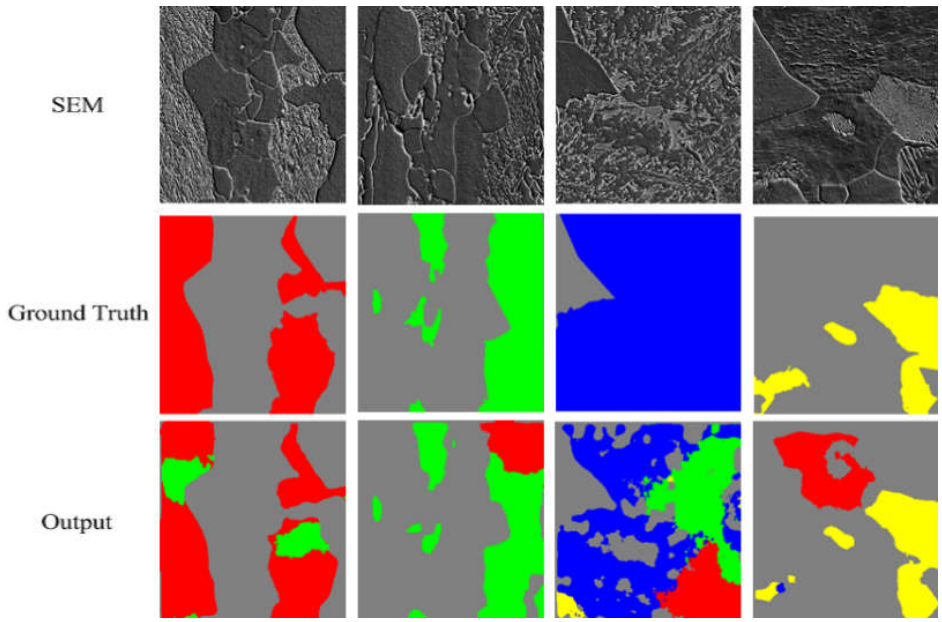
Network	Data	Balanced	Augmented	Training Strategy	pixel acc.	mean acc.	Mean IU	f.w. IU
FCNN	SEM	-	-	scratch	80.84	33.99	24.25	71.74
FCNN	SEM	-	-	fine tuning	92.01	76.26	68.26	86.80
FCNN	SEM	✓	-	fine tuning	92.11	79.63	66.27	86.91
FCNN	SEM	✓	✓	fine tuning	**93.92**	**76.70**	**67.84**	**88.81**
FCNN	LOM	✓	✓	fine tuning	88.27	54.01	45.43	81.66

6 Discussion

Classification using object-based CNNs: Based on Table 1, all of the Deep Learning methods outperform the state-of-the-art method which proves our motivation regarding using learned high-level features rather than engineered features. The results also indicate that deeper networks show better results than shallower ones i.e. depth matters. Even with features extracted from pre-trained VGG19, which are classified with SVM, one can achieve comparable performances with the results of trained CIFARNET and VGG16. However, training VGG16 on the dataset used in this work makes features more informative and discriminative as network parameters can learn the pattern in this dataset. The surprising performance of the MVFCNN method indicates that the performance of object-based CNNs is negatively influenced by the image resizing step. In other words, we observe that resizing distorts the texture inside objects, hampering the accurate differentiation of objects bigger or smaller than the input size of the network. In another approach, we split big objects into 224x224 px objects which led to a higher performance which supports our assumption. However, this approach made the system less practical and introduces a hyper



(a) *Success cases*



(b) *Failure cases*

Figure 6. *a)Examples of a)successful and b)failure cases in SEM segmentation using the best MVFCNN approach configuration. The ground truth colors of martensite, tempered martensite, bainite and pearlite are red, green, blue and yellow respectively.*

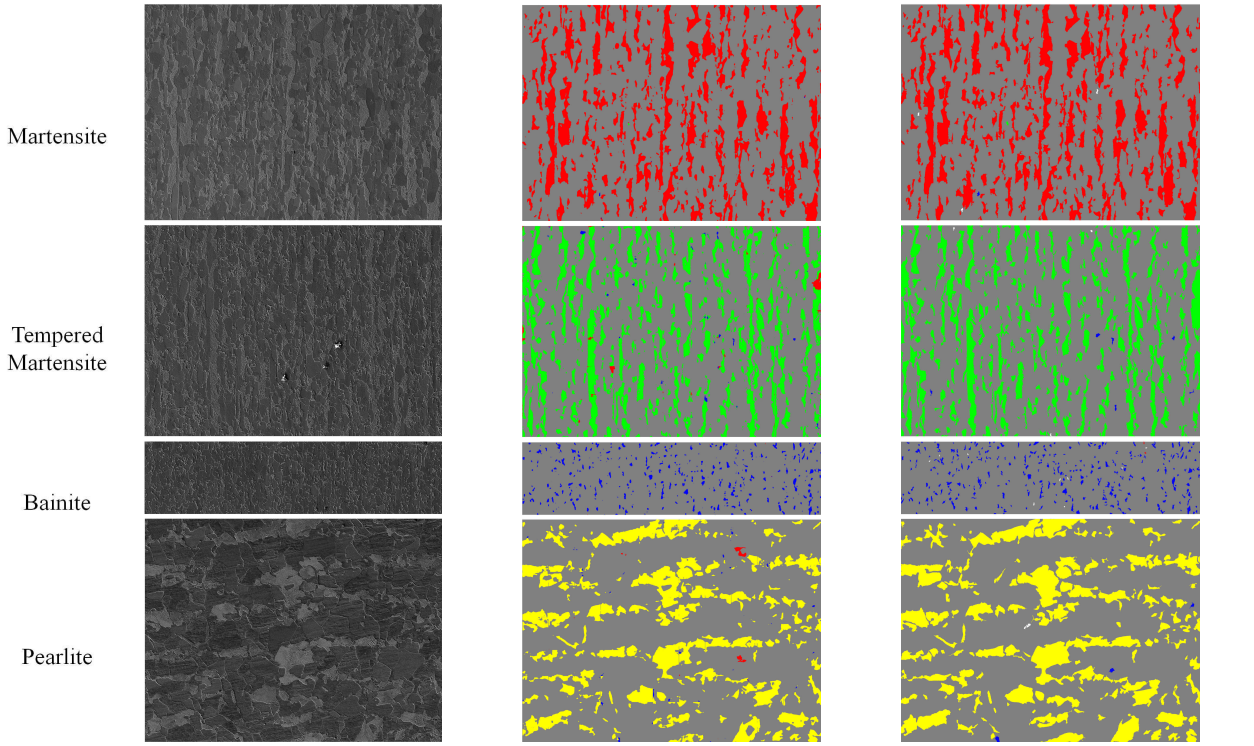
parameter into the system to choose the best split size.

Classification using MVFCNNs: The results in Table 1 shows that a MVFCNN approach using SEM images achieves a very high performance. It indicates that a classification using pixel-wise segmentation is more efficient and accurate compared to object-based CNN methods and significantly better than hand-crafted features. Resizing is no longer required in this method and therefore does not have an impact on the performance of the MVFCNN.

The confusion matrix in Table 4 shows that the network still produces some misclassification of martensite objects due to the confusion of martensite with tempered martensite and bainite which have similar textures – unlike the texture of pearlite's texture which is easy to distinguish. All wrongly classified bainite objects belong to the "transformed" bainite sample in the test set which is not present in the training set. It is impressive that in this condition the network is still able to classify more than

Table 6. *Pixel-wise accuracy evaluation of the segmentation approach using FCNNs for each microstructure class. The matrix has the highest and bainite the lowest pixel-wise accuracy.*

Network	Data	Balanced	Augmented	Training Strategy	Matrix	Marten.	Temp. Marten.	Bain.	Pear.
FCNN	SEM	-	-	scratch	86.47	36.08	0	0	0
FCNN	SEM	-	-	fine tuning	92.03	77.25	71.43	22.05	63.27
FCNN	SEM	✓	-	fine tuning	92.47	74.71	57.44	33.25	69.65
FCNN	SEM	✓	✓	fine tuning	**94.22**	**79.85**	**72.62**	**37.32**	**70.46**
FCNN	LOM	✓	✓	fine tuning	94.11	50.11	58.36	5.21	19.35



(a) *Original images* ***(b)*** *Stitched segmented patches* ***(c)*** *Classification after max-voting*

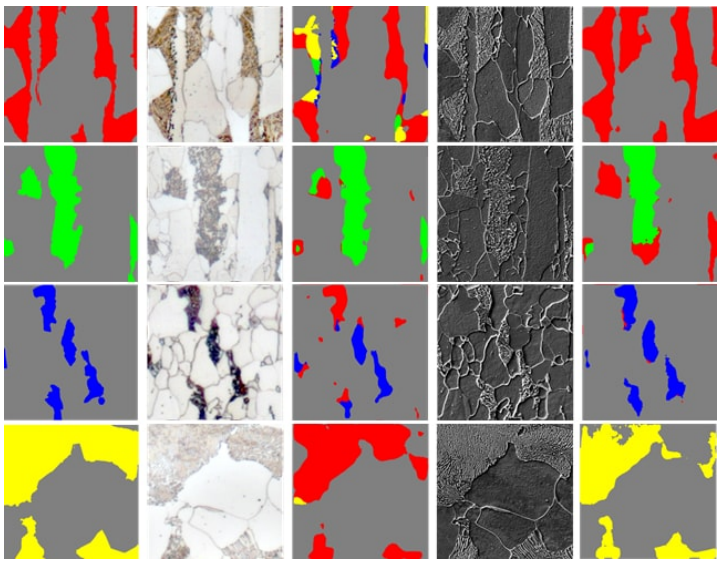
Figure 7. *Examples of applying a max-voting scheme to stitched and segmented SEM patches for different microstructure classes. The ground truth colors of martensite, tempered martensite, bainite and pearlite are red, green, blue, and yellow, respectively. Not-segmented objects in the segmentation phase are shown with white color.*

half of the objects in the "transformed" bainite sample correctly. The achieved high accuracy indicates that considering each pixel and taking into account its neighboring pixels plays a crucial role in the correct classification of objects.

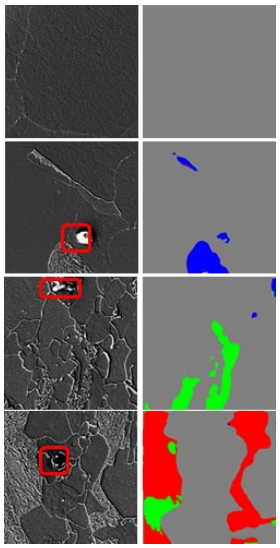
Based on the results of Table 2, data augmentation improves the performance by 2 percent points which is not considerable. One possible reason for this phenomenon is that different rotations of textures inside objects are already present in the dataset before data augmentation which the network has already seen and learned. For example, in the pearlite microstructure, there are many cases that by rotating a patch, the resulting augmented patch still contains the previous orientations before the rotation.

The results presented in Table 3 confirm our expectation that LOM images will perform poorly compared to SEM due to lower resolution. The artifacts due to stitching as well as the different illumination and preparation residues and scratches in LOM images degrade classification accuracy. The results in Table 3 have the same trend compared with Table 1 and confirm our findings. Table 5 shows that using pixel-wise classification using MVFCNNs, one can achieve a high performance of 93.94% accuracy despite of more complex microstructural textures. These results also show that the better the pixel-wise segmentation criteria are, the better the microstructural classification will be.

Regarding Table 6, the results show how important fine tuning is when working with little training data. Without fine tuning, tempered martensite, bainite and pearlite microstructures could have not been segmented at all. Low performance in bainite



(a) *Comparison of SEM and LOM pixel-wise semantic segmentation*



(b) *Model robustness to noise*

Figure 8. *(a) Comparison of LOM and SEM segmentation using FCNN network. Columns from left to right: Ground truth, LOM, LOM segmentation, SEM and SEM segmentation. (b) Noisy patches segmentation. First top row shows ferrite and the rests is noise. The ground truth colors of martensite, tempered martensite, bainite and pearlite are red, green, blue, and yellow, respectively.*

class even by doing data augmentation could be due to the fact that bainite objects compared to other classes are a lot smaller. In the dataset, there are quite a few tiny areas belonging to matrix class which are not completely plain and contain some textures similar to those in bainite. As as result, balancing and augmenting the dataset makes the network decide that those small objects are more likely matrix than bainite. This assumption is verified by observing the bainite test image that pixels are either classified as bainite or as matrix. Another reason is also presence of isothermally transformed bainite which has different appearance with bainite of the other sample and network was not trained on that. More training data, specially for bainite class according to [34] would help to decrease the miss-classified bainite objects in the segmentation step. This is in process for the next generation although it is very hard to find enough samples with a sufficient number of objects - especially for upper and lower bainite - which can be separated from a matrix to define the ground truth and to have a sufficiently good training dataset. Moreover, first efforts are heading in the direction of producing lab melts which are then analyzed by dilatometry to further enhance the division of the classes. Also the ferrite including its grain boundaries will be added in a next iteration as additional class. Regarding noise robustness, Figure 8b shows strong robustness of the system with different types of noises. We also noted that the system is rotation invariant which means similar patches with different rotation angles have the same system response. It is worth mentioning that MVFCNNs can also have input images smaller than 1000x1000 px. However, in input image smaller than 100x100 px the response is for the same area worse than has 1000x1000 px.

7 Conclusion

This work demonstrates the feasibility of an effective steel microstructural classification using Deep Learning methods without a need of separate segmentation and feature extraction. We performed a pixel-wise microstructural segmentation using a trained FCNN network followed by a max-voting scheme. The observed strong improvements in classification performance over the prior state of the art confirm our idea of leveraging the raw data input for training Deep Learning-based classification systems. Besides the high accuracy result, we are able to achieve a very fast prediction.

We found that resizing objects directly as input to object-based CNNs can eliminate discriminative texture information relevant to different microstructural classes. In contrast, MVFCNN approach does not have this problem and it is independent of the size of objects. Data augmentation was considered for further performance improvements. Furthermore, we found that rotating the SEM images does not introduce considerable new information and therefore the performance is not significantly improved. We conclude that pixel-wise segmentation using Fully Convolutional Neural Networks is an effective and robust way of determining the distribution and size of different microstructures when these networks are trained end-to-end.

8 Acknowledgements

We are grateful to NVIDIA company for granting us with a GPU.

9 Additional information

The authors declare no competing financial interests.

References

1. Tasan, C. C. *et al.* An overview of Dual-phase steels: Advances in Microstructure-oriented Processing and Micromechanically Guided Design. *Annu. Rev. Mater. Res., vol. 45, no. 1* 391–431 (2014).

2. Khedkar, P., Motagi, R., Mahajan, P. & Makwana, G. A Review on Advance High Strength Steels. *Int. J. Curr. Eng. Technol., vol. Special Is, no. 6* 240–243 (2016).

3. Bhadeshia, H. & Honeycombe, R. Steels: Microstructure and Properties. *Elesevier - Butterworth-Heinemann - 4* (2017).

4. Ohser, J. & Muecklich, F. Statistical Analysis of Microstructures in Materials Science. *John Wiley & Sons J.* (2000).

5. Aarnts, M. P. Microstructural quantification of multi-phase steels (micro-quant). *Eur. Comission, Luxembourg, DOI: 10.2777/83656* (2011).

6. Krauss, G. Steels: Processing, Structure, and Performance. *ASM Intl. 2* (2015).

7. Pereloma, E. & Edmonds, D. V. Phase Transformations in Steels: Diffusionless Transformations, High Strength Steels, Modelling and Advanced Analytical Techniques. *Woodheard, Camb.* (2012).

8. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-scale Image Recognition. In *International Conference on Learning Representations (ICLR)* (2015).

9. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998).

10. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J. & Fricout, G. Steel Defect Classification with Max-pooling Convolutional Neural Networks. *Proc. Int. Jt. Conf. Neural Networks* (2012).

11. Vander Voort, G. Metallography: Principles and Practice. *McGraw-Hill* (1984).

12. Beraha, E. & Shpigler, B. Color Metallography. *Am. Soc. for Met.* (1977).

13. Shrestha, S. L. *et al.* An Automated Method of Quantifying Ferrite Microstructures Using Electron Backscatter Diffraction (EBSD) Data. *Ultramicroscopy Journal, 137,* 40–47 (2014).

14. Bhadeshia, H. & Honeycombe, R. Steels: Microstructure and Properties. *Elsevier Ltd. J.* (2006).

15. Gerdemann, F. Bainite in Medium Carbon Steels. *Shak. Verlag J.* (2010).

16. Friel, J. J. Practical Guide to Image Analysis. In *ASM Intl.* (2000).

17. Britz, D., Webel, J., Schneider, A. & Mücklich, F. Identifying And Quantifying Microstructures in Low-alloyed Steels: A Correlative Approach. *Metall. Italiana, 3* 5–10 (2017).

18. Britz, D., Hegetschweiler, A., Roberts, M. & Mücklich, F. Reproducible Surface Contrasting And Orientation Correlation of Low Carbon Steels By Time Resolved beraha Color Etching. *Mater. Perform. Charact. Vol. 5,* 553–563 (2016).

19. Velichko, A. Quantitative 3d Characterization of Graphite Morphologies in Cast Iron using fib Microstructure Tomography. *Saarbruecken, Shak. Verlag* (2009).

20. Drucker, H., Burges, C., Kaufman, L., Smola, A. & Vapnik, V. Support vector regression machines. In *Neural Information Processing Systems (NIPS)* (1996).

21. Pauly, J., Britz, D. & Mücklich, F. Advanced Microstructure Classification Using Data Mining Methods. In *TMP - 5th International Conference on ThermoMechanical Processing* (2016).

22. Krizhevsky, A., Ilya, S. & Hinton, G. Imagenet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems (NIPS)* (2012).

23. Deng, J. *et al.* Imagenet: A Large-scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

24. Long, J., Shelhamer, E. & Darrel, T. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

25. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

26. Cordts, M. *et al.* The cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

27. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Neurocomputing: Foundations of research. chap. Learning Representations by Back-propagating Errors, 696–699 (MIT Press, Cambridge, MA, USA, 1988).

28. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In *Biological Cybernetics* (1980).

29. Srivastava, R., Greff, K. & Schmidhuber, J. Training Very Deep Networks. *Neural Inf. Process. Syst. (NIPS)* (2015).

30. Donahue, J. *et al.* DeCAF: A Deep Convolutional Activation Feature for Generic Visual recognition. In *Journal of Machine Learning Research (JMLR)* (2014).

31. Britz, D., Webel, J., Gola, J. & Mücklich, F. A Correlative Approach to Capture and Quantify Substructures by Means of Image Registration. *Pract. Metallogr. Vol. 54, No. 10* 685–696 (2017).

32. Dalal, N. & Triggs, B. Histograms of Oriented Gradients for Human Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005).

33. Jia, Y. *et al.* Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, 675–678 (ACM, 2014). URL http://caffe.berkeleyvision.org.

34. Zajac, S., Schwinn, V. & Tacke, K. H. Characterisation and Quantification of Complex Bainitic Microstructures in High and Ultra-high Strength Linepipe Steels. *Mater. Sci. Forum* 500–501 (2009).