

## Autonomous Interpretation of the Microstructure of Steels and Special Alloys

MULEWICZ Bartłomiej<sup>1,a\*</sup>, KORPALA Grzegorz<sup>2,b</sup>, KUSIAK Jan<sup>1,c</sup>  
 and PRAHL Ulrich<sup>2,d</sup>

<sup>1</sup>AGH University of Science and Technology, Kraków, Poland

<sup>2</sup>TU Bergakademie, Freiberg, Germany

<sup>a</sup>bartekmulewicz@gmail.com, <sup>b</sup>Grzegorz.Korpala@imf.tu-freiberg.de, <sup>c</sup>kusiak@agh.edu.pl,

<sup>d</sup>Ulrich.Prahl@imf.tu-freiberg.de

**Keywords:** Microstructure analysis, Artificial Intelligence, Deep Learning.

**Abstract.** The main objective of presented research is an attempt of application of techniques taken from a dynamically developing field of image analysis based on Artificial Intelligence, particularly on Deep Learning, in classification of steel microstructures. Our research focused on developing and implementation of Deep Convolutional Neural Networks (DCNN) for classification of different types of steel microstructure photographs received from the light microscopy at the TU Bergakademie, Freiberg. First, brief presentation of the idea of the system based on DCNN is given. Next, the results of tests of developed classification system on 8 different types (classes) of microstructure of the following different steel grades: C15, C45, C60, C80, V33, X70 and carbide free steel. The DCNN based classification systems require numerous training data and the system accuracy strongly depend on the size of these data. Therefore, created data set of numerous micrograph images of different types of microstructure (33283 photographs) gave the opportunity to develop high precision classification systems and segmentation routines, reaching the accuracy of 99.8%. Presented results confirm, that DCNN can be a useful tool in microstructure classification.

## Introduction

The effort of promoting light weight design in the wide spectrum of industry can be realized by application of new materials. There are two groups of materials which are being focused on by industry at the moment. One group consists of materials with a low density such as aluminum, magnesium or carbon fiber reinforced polymers, while the other group consists of high strength steels. The latter group includes multiphase steels which offer, besides high strength, a good forming ability. Another advantage of these steels can be found in the low alloy element content which often is below 5%, not leading to an increase of costs.

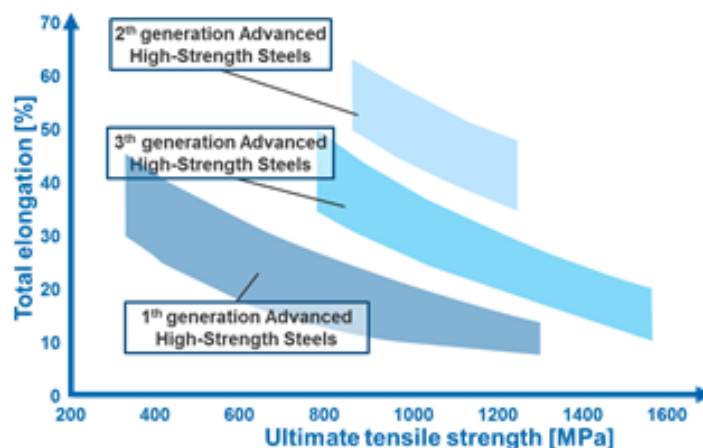


Figure 1: Division of modern light weight steels (AHSS)

Advanced High Strength Steels (AHSS) can be divided into three generations, while the most popular is the 1st generation of AHS steels. Figure 1 shows different mechanical properties of steels of all three AHSS generations. For so large spectrum of mechanical properties there are five hardening phenomena which result in the strength value (see Table 1). The most frequently used in controlling the mechanical properties, next to precipitation hardening, is the phase hardening which can be easily modified in production chain. This is controlled by the composition of phases in microstructure and can be modified by change of carbon content and cooling strategy.

Table 1: Hardening phenomena according to Hougardy [1]

Hardening Phenomenon	Yield strength [MPa]	Toughness
Solid solution strengthening	500	↓
Precipitation hardening	3000	↓
Grain boundary strengthening	500	↑
Strain hardening	1500	↓
Phase hardening	3000	↓
Theoretical Max. Hardening	8500	
Now produced steels	Up to 2500 for strip Up to 4000 for wire	

Moreover, heat treatment condition as shown in Figure 1, give insight how to transform austenite to generate targeted microstructure. A combination of different microstructure types: martensite, bainite, perlite and/or ferrite (in order - from the hardest to the softest one) results in different mechanical properties.

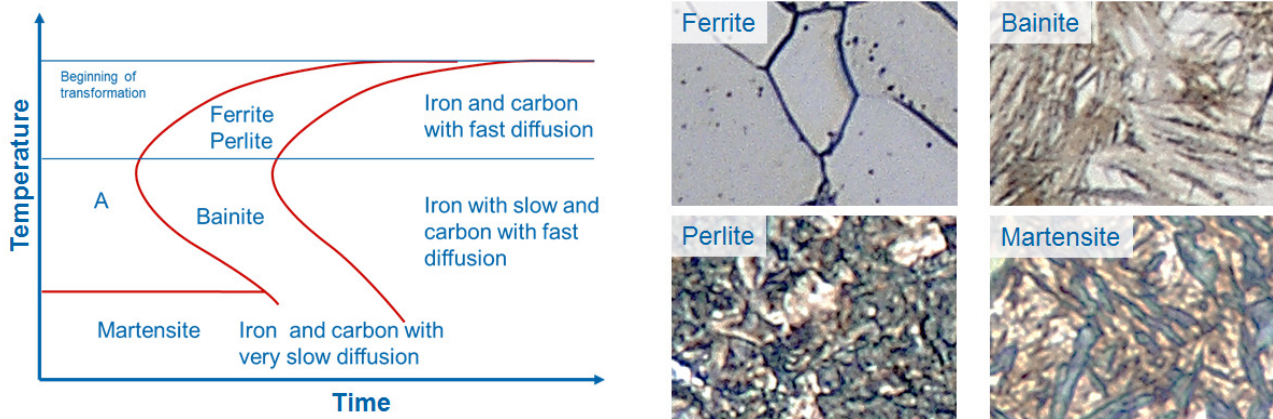


Figure 2: Time-temperature-transformation diagram and major transformation products in steel

While microstructural characterization is very important and pretty well recognized, its classification is not a trivial task. Despite the dynamic development of digital photography and computer systems, classification of the steel microstructure remains the task of experts who "manually" evaluate a given picture of the structure. Classification appears extremely difficult especially in the cases, when there are mixtures of different phases with various substructures. There is no evidence of the computer systems which allow the automatic classification of microstructure, so any attempt in that direction can be valuable.

The main goal of our research is to develop a system for microstructure image recognition based on the Artificial Neural Networks, specifically on the Deep Learning. Deep Convolutional Neural Networks become a useful tool in the image recognition in the pretty all domains [2, 3, 4, 5].

Examples of attempts of application of deep CNNs in the materials science can be found in work [6], where deep CNN was used in detection of defects on upper and lower part of rolled steel strips. Other examples of deep CNNs applications to the microstructure analysis and classification can be found in [7, 8]. Authors of work [7] uses deep CNNs only as a feature extractor and do not fully capitalize on its potential. Second approach is very similar to our approach in algorithmic level but is not fully compatible with analyzed data set and thus achieved results probably can be improved. Moreover, we used some additional techniques from recent progress in computer vision field that should further improve obtained results, as they did in other image analysis tasks.

The developed DCNN based classification system was validated firstly on the data of work [7] and tested next on 8 different types (classes) of microstructure (see Table 2) of the following different steel grades: C15, C45, C60, C80, V33, X70 and carbide free steel.

## Deep Convolutional Neural Networks

**Artificial Neural Networks.** Artificial Neural Networks (ANNs), in their classical form, consist of fully connected layers that are often called dense layers (Fig. 3). Each layer can be associated with a vector of numbers, called neurons. Input vector corresponds to the first layer and the model output to the last layer. Values of neurons in the next layer are calculated from values of previous layer according to the formula:

$$H_n = f(H_{n-1} * W_n + b_n)$$

where  $H_{n-1}$  is a vector of values from the previous layer,  $W_n$  is a matrix which is usually adjusted during training process for better fitting to the data. Similar mechanism works for a vector of real values  $b_n$ , and  $f$  is so called activation function that works element wise on each element [9]. The function  $f$  is usually selected from a known set of functions, currently the common choice is  $f(\mathbf{x}) = \max\{0, \mathbf{x}\}$ , called ReLU (Rectified Linear Unit [9]).

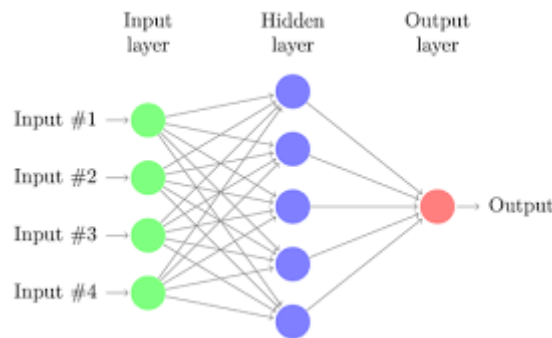


Figure 3: Classical neural network

Most effective way to train neural network is the back-propagation method with gradient descent optimization procedure. Training step (optimization procedure) modifies neural network model parameters by small amounts minimizing value of loss function  $L$ , which assesses model performance [10]. Because of large amounts of data, stochastic version of this method [11, 12] is used in almost all cases. The only difference is in the fact that derivatives of  $L$  and consequently updates of parameters are calculated for some randomly selected part of the data, not for all records. The procedure is conducted until all the data is considered and then repeated. One run through all available data is called epoch. Learning process can take from one to thousands of epochs, depending on problem complexity, amount of available data, and computation power that is available.

**Convolutional Neural Networks.** Convolutional Neural Networks (CNN) were firstly presented in [13] and are based on experiments on how human visual cortex works [14]. They are composed of convolutional layers, which base on convolutional filters and activation functions (Fig. 4). The major difference between them and dense layers is that each value is calculated only using small part of data

from previous layer, corresponding to the surrounding. We call that part of data a slice. We obtain the slice by using a filter, which is a small sized tensor. One layer can be composed of any number of filters of the same size. In the simplest type of convolutional layer, we select part of the data which have the same size as filter and then apply the convolution operation to these two vectors and add bias to that value, by analogy to a dense layer. We repeat this procedure to every possible slice of the data that has the same size as filter. We also have to preserve the spatial nature of the data, so outputs of two adjacent slices are also adjacent in the same way.

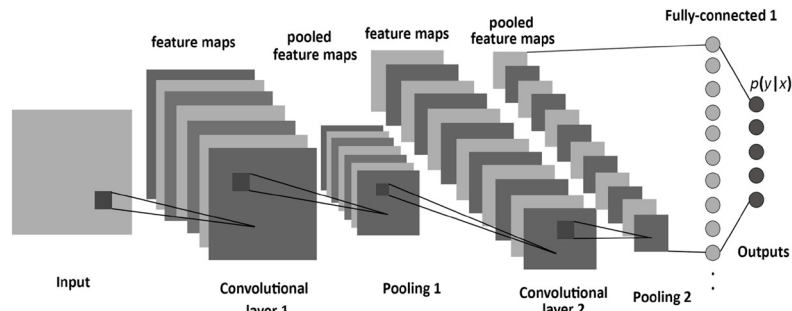


Figure 4: Convolutional Neural Network [15]

First time when CNNs achieved extraordinary results was for MNIST (Modified National Institute of Standards and Technology) data in 1998 [16]. It had 3 convolutional layers and just one dense layer at the end. Right now, there is a real renaissance of this technique, since 2012 when AlexNet [17] and ImageNet [18] were launched. That CNN architecture was composed of 5 convolutional and 3 dense layers. Since then, they are achieving better and better results in computer vision tasks such as image recognition [11 12, 13], segmentation, detection, video description and others. They also work very well in other fields in which structure of the data is very obvious and important [19, 20].

**Deep Convolutional Neural Networks.** Deep Convolutional Neural Networks (DCNNs) achieve significantly better results mostly because they have much more parameters than traditional neural networks but they require huge amounts of data [2]. Fortunately, nowadays big databases with a lot of examples are widely available, as it was in our case of the microstructure photographs. In this paragraph we briefly describe methods that were implemented in our DCNN approach and our learning algorithm. We provide references for more inquisitive readers.

Stochastic Gradient Descent (SGD) algorithm is one of the easiest ways to train DCNN. However, training of neural networks using the SGD takes a lot of time in case of so vast data. Therefore, we used SGD with nesterov momentum to speed up training [21, 22].

Another innovation which can improve CNN is the idea of implementation of pooling layers to the CNNs [23, 2, 3]. In our model, we used max pooling layer, which divides input into smaller pieces and from each piece extracts the maximal value. This trick decreases dimension of hidden layers. For example, if we divide data from each filter into 2x2 squares, the size of output from pooling layer is two times smaller in each of the first two dimensions. It works as a regularizing mechanism and improves performance in case of new data, and helps to avoid the unwanted overfitting.

Next modification that we used has been presented in ImageNet 2015 and is called residual connection [4]. The idea is to add outputs from  $i$ -th layer to  $(i+k)$ -th layer, where  $k$  is usually small in comparison to number of layers in the whole network. It allows to build DNNs with thousands of layers, which does not increase the computation time over the acceptable limit. It is because the layers between beginning and end of residual connection more adjust transferred values than learn new features. It also fixes the vanishing gradient problem, that until then was still an issue for NN with multiple layers. Architecture with this connection defeated lately the ImageNet and also for the first time achieved better image classification accuracy than a human.

Batch normalization [24] is another method that is widely used to speed up training process. It can be applied to the output of any layer. We normalize variance to 1 and mean to 0 and recalculate new mean and variance, to achieve better performance.



DCNNs work best with very big data sets, with thousands of examples per class. When such amount of data is not available one can apply transfer learning [25]. In this technique we take DNN trained on a big data set that achieved very good result and slightly modify its architecture to solve new problem with much smaller amount of data. Pretrained DNNs are publicly available in python libraries. We used ResNet18 architecture that was trained on ImageNet. Its architecture is presented in [4].

### Experimental Setting

The main goal of our research was to develop and implement DCNN for classification of microstructure types based on the light microscopy photographs. To get image datasets for this task, a heat treatment was performed in dilatometer for different chemical compositions of steel. Dilatometer samples with a diameter of 5 mm and a length of 10 mm were manufactured from the wire, and dilatometric investigations were carried out on a DIL 805 dilatometer from TA Instruments. The samples were austenized at the temperature of 1000 °C, and heat treated as follows:

Table 2: Microstructure types and the heat treatment conditions

Class	Microstructure type	Steel	Heat treatment
0	bainite with carbide	C15	cooling rate 50 K/s
1	granular bainite	X70	cooling rate 50 K/s
2	columnar bainite	X70	cooling rate 30 K/s
3	carbide free bainite	steel from [26]	Isothermal transformation by 350 °C
4	ferrite with grain boundaries	electro steel V33	cooling rate 1 K/s
5	ferrite with grain boundaries and small islands of perlite	C15	cooling rate 1 K/s
6	martensite	C60	cooling rate 100 K/s
7	perlite	C80	cooling rate 1 K/s

After the heat treatment of the samples, their microstructure was analysed. Photographs of the microstructure were made using optical digital microscope Keyence VHX 6000. All images were stored in the database on the server, and used next as the DCNN training and testing data.

All experimental work of heat treatment of steel and acquisition of photographs of microstructures were performed in the laboratory at the TU Bergakademie, Freiberg, while development and implementation of DCNN system were done at AGH UST, Kraków.

Images, that were originally used for training in ImageNet were of size 224x224x3 and our images were 1200x1600x3. Because convolutional layers are sliding through input there is no problem to slide them for a bigger image. The problem is in connection from convolutional layer to dense layer, as for dense layer we need to have connection from each input to each output. We could either slice input into smaller pieces of 224x224, or to decrease the dimension of input using techniques from computer vision, or else to change DCNN architecture to allow higher dimensional pictures. We decided to choose the third option, but not in the most obvious way. In original ResNet18 network trained on ImageNet data, output from last convolutional layer had size 7x7x512 and first dense layer had 4096 neurons, which yields connections with over  $10^8$  parameters. In case of the image of size 1200x1600x3, the output size is 36x48x512, so for connection with dense layer with 4096 neurons we would need over  $3.6 * 10^9$  parameters, 36 times more than in initial network. That many parameters require many GPGPUs for training, which risks of overfitting in case of our relatively small data set. To deal with this problem, we decided to use approach from object segmentation idea, in which the image is divided into separate classes, pixel by pixel [27]. We changed pretrained architecture replacing the last dense layers with convolutional ones with filter of the size of 1 in first

two dimensions, creating so called Fully Convolutional Network (FCNN) [5] with number of filters in the last layer corresponding to the number of classes in the data set. This way, each output has a size equal to *number of classes*  $\times$   $36 \times 48$ , and a tensor of size *number of classes*  $\times$   $1 \times 1$  can be understood as evaluation of classification for small part of input image. Then we applied *softmax* to evaluate a mean value of every class activation, calculated from  $36 \times 48 = 1728$  values. This was also taken into account in backpropagation procedure. Our output is a mean value of outputs for consecutive parts of image. This significantly improve a model accuracy and it is also proven that such approach appears better than applying the network separately to consecutive small parts of the input [5]. This approach can't be usually applied in classification problem, because the picture of the microstructure type does not cover the whole picture and is context dependent. However, we could apply it because each photograph in our input data set represents a single type of microstructure.

## Results

Described model was run for the microstructure photographs of 8 classes of microstructure listed in the Table 2. Each photo represented a single microstructure class, however, due to different conditions, some of them are very dissimilar. All the photographs were preprocessed. In the case the photo background was more than a half of the considered image, this image was automatically excluded. Also, the blurred images were eliminated. From remaining data, we selected 1000 or 2000 for training and the rest was left as a testing data set. The developed DCNN based model achieved 99,88% accuracy (percentage of properly classified examples) for analyzed training data set, misclassifying 15 out of 13000 images (most of them were blurred). For the testing data which did not participate in the training process, the accuracy was 96.47% (715 out of 20283 were misclassified). Details of obtained results are presented in Table 3. Most of misclassified examples are blurred. This applies especially to all misclassified images from class 3 (carbide free bainite).

Table 3: Results of classification of the testing data set

True/Predicted	0	1	2	3	4	5	6	7	Total
0	284	0	0	0	0	0	0	0	284
1	12	7200	10	2	1	1	0	0	7226
2	37	0	4230	0	0	2	0	0	4269
3	645	0	0	3670	0	0	0	0	4315
4	3	0	0	0	527	0	0	0	530
5	0	0	0	0	0	543	0	0	543
6	0	0	0	0	0	0	265	0	265
7	1	0	0	1	0	0	0	2849	2851
Total	982	7200	4240	3673	528	546	265	2849	20283

## Summary

Presented results show that the elaborated Deep Convolution Neural Network based system can be used in classification of different types of microstructure of low alloy steels. Obtained accuracy is very promising.

Other works that tried to use Deep Learning for microstructure classification have some important drawbacks. In work [7] the neural network was used only as a feature extractor. Second drawback of [7] is that experiments were performed on small dataset (600 images from 3 classes) of images of poor quality and diversified magnification. To validate our system, we reused these data of work [7]. Achieved results are slightly better (however not significantly).

Our future research will focus on developing models that will be able to discriminate more and more types of microstructure types. We also consider using some transformations that help humans in the microstructure classification and recognition. Next, we want to solve segmentation problem, in which we have different microstructures in one image and want to classify them. Our Fully Convolutional Network trained for classification will definitely be used in that problem, as it is state-of-the-art solution right now. Therefore, the Authors plan to continue research in this domain, as well as in application to other, similar problems of materials engineering.

### Acknowledgements

Financial assistance in frame of the AGH project No 11.11.110.593 is acknowledged.

### References

- [1] H. P. Hougardy, Hans Paul. Umwandlung und Gefüge unlegierter Stähle. Düsseldorf: Stahleisen GmbH, 2003. 3-514-00423-4.
- [2] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
- [3] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [5] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [6] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber and G. Fricout, "Steel defect classification with Max-Pooling Convolutional Neural Networks," The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, 2012, pp. 1-6. doi: 10.1109/IJCNN.2012.6252468.
- [7] Brian L. DeCost, Toby Francis, Elizabeth A. Holm, Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. Acta Materialia, 133, (2017), 30-40, doi: 10.1016/j.actamat.2017.05.014.
- [8] Azimi, Seyedmajid, et al (2018). Advanced Steel Microstructural Classification by Deep Learning Methods. Scientific Reports. 8. 10.1038/s41598-018-20037-5.
- [9] I. Goodfellow, et al. Deep learning. Vol. 1. MIT press, Cambridge, 2016.
- [10] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533.
- [11] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." Proceedings of COMPSTAT2010. Physica-Verlag HD, 2010. 177-186.
- [12] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.
- [13] Fukushima, Kunihiro, and Sei Miyake. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition." Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982. 267-285.
- [14] Hubel, David H., and Torsten N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex." The Journal of physiology 195.1 (1968): 215-243.
- [15] Albelwi, S.; Mahmood, A. A Framework for Designing the Architectures of Deep Convolutional Neural Networks. Entropy 2017, 19, 242.

- 
- [16] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
- [17] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.
- [18] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015): 211-252.
- [19] Borovykh, Anastasia, Sander Bohte, and Cornelis W. Oosterlee. "Conditional time series forecasting with convolutional neural networks." *arXiv preprint arXiv:1703.04691* (2017).
- [20] Mulewicz, Bartłomiej, et al. "Failures prediction based on performance monitoring of a gas turbine: a binary classification approach." *Schedae Informaticae* 26.9 (2017): 21.
- [21] Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2013.
- [22] Qian, Ning. "On the momentum term in gradient descent learning algorithms." *Neural networks* 12.1 (1999): 145-151.
- [23] Goodfellow, Ian J., et al. "Maxout networks." *arXiv preprint arXiv:1302.4389* (2013).
- [24] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).
- [25] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.
- [26] G. Korpała, PhD thesis, Gefügeausbildung und mechanische Eigenschaften von unlegiertem bainitischem Warmband mit Restaustenit, 2017
- [27] Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88.2 (2010): 303-338.