Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVII, págs. 2023-2032 (artículo completo)
A. Cardona, L. Garelli, J.M. Gimenez, P.A. Kler, S. Márquez Damián, M.A. Storti (Eds.)
Santa Fe, 5-7 Noviembre 2019

AUTOMATIC STEEL MICROSTRUCTURAL QUANTIFICATION BY CONVOLUTIONAL NEURAL NETWORKS

Cássio Danelon[a], Thales Tozatto[a], Leonardo Goliatt[a], Moisés Lagares[b], Gulliver Catão[b] and Lecino Caldeira[c]

[a]*Graduate Program in Computational Modeling, Federal University of Juiz de Fora, Brazil, https://www.ufjf.br/pgmc*

[b]*Dept. of Mechanical Engineering, Federal University of Juiz de Fora, Brazil, www.ufjf.br/mecanica/*

[c]*Dept. of Metallurgy, Federal Institute of Education, Science and Technology of Minas Gerais, Brazil, https://www.ifsudestemg.edu.br/juizdefora*

Keywords: Microstructural analysis, Computer vision, Convolutional neural networks.

Abstract. The microstructural analysis of a material allows the complete characterization of its mechanical properties. Thus, the performance of a mechanical component depends heavily on the identification and quantification of its microstructural constituents. Currently, this process is still done mostly manually by experts, making it slow, very labor-intensive and inefficient. It is estimated that an experienced expert takes 15 minutes per image to perform the proper identification and quantification of microconstituents. Therefore, a computational tool could greatly assist to improve the performance in this task. However, since a microstructure can be a combination of different phases or constituents with complex substructures, their automatic quantification can be very hard and, as a result, there are few previous works dealing with this problem. Convolutional Neural Networks are promising for this type of application since recently this type of network has achieved great performance in complex applications of computational vision. In this work, we propose an automatic quantification of microstructural constituents of low carbon steel via Convolutional Neural Networks. Our dataset consists of 210 micrographs of low carbon steel, and this amount of images was increased through data augmentation techniques, resulting in a total of 672 samples for training. With regard to network architectures, we used the AlexNet trained from scratch and the VGG19 and Xception both pre-trained. The results showed that CNNs can quantify microstructures very effectively.

1 INTRODUCTION

The mechanical properties of a material are determined essentially by the analysis of its microstructures (Bhadeshia and Honeycombe, 2017). These microstructures stores the genesis of a material and determines all its physical and chemical properties (Azimi et al., 2018). Hence, the identification and quantification of microstructural constituents of a material is a critical aspect in engineering.

Yet, even today, there is no automatic way of quantifying the microstructural constituents of a material. This task has to be done manually by experts or with the aid of expensive and inefficient devices. The American Society for Testing and Materials (ASTM) produces industrially recognized standards for the analysis of microstructure evaluated by human experts. However, implementing these standards is labor intensive and prone to human error (Campbell et al., 2018).

This motivates us to use Deep Learning (DL) methods to automatically quantify these microconstituents in a material. These machine learning methods are recently receiving the attention of scientists due to their strong potential to learn complex features. In recent years, due to its remarkable precision in images patterns recognition and classification, Convolutional Neural Networks (CNNs) are considered the most suitable DL method to treat with computer vision tasks. There are some works that use CNNs to classify microstructures. However, the works that use this tool to quantify the microconstituents are rare.

In this work, CNNs was applied to quantify five very common microstructures in welded fusion zones: Grain Boundary Ferrite (PF(G)), Intragranular Polygonal Ferrite (PF(I)), Ferrite with Aligned Second Phase (FS(A)), Ferrite with Non-aligned Second Phase (FS(NA)) and Acicular Ferrite (AF). Three different CNNs architectures were compared: AlexNet trained from scratch and the VGG19 and Xception both pre-trained.

2 RELATED WORKS

In the literature, it is possible to find some papers that aim at the automatic classification and analysis of microstructures. Deep Learning methods were used by Azimi et al. (2018) to automatically classify microstructures, such as martensite, bainite, and perlite.It was used different CNNs architectures and also Support Vector Machines (SVM). The best results (93.94%) were obtained using the MVCNN (Max-voting Fully Convolutional Neural Network).

Gola et al. (2018) classified microstructures of two distinct steels by data mining methods. The results were obtained from different methods of preprocessing (raw and log-transformed), different methods of division of data (shuffled and sample-wise) and different parameters C and gamma of the SVM method. The best results reached 87% accuracy.

Supervised and unsupervised Machine Learning techniques were used by DeCost et al. (2017) for the identification of microstructures in thermally treated high carbon steels. Different feature extractors were used, e.g CNN and Bag of Words, and the learning algorithm SVM as classifier. The results showed that CNNs were the best feature extractor. For the visualization maps, the t-SNE clustering technique was used.

Specific microstructures with different magnitudes, chemical treatments and orientations were automatically identified by Chowdhury et al. (2016). Two classification tasks were developed: classification between micrographs with and without dendrites, and classification between views of longitudinal dendrites and cross-sections. The learning algorithms SVM, KNN, and Random Forests were used in the classification task. The best results were obtained when pre-trained neural networks were used to extract the images features.

3 METHODS

In this section, we describe our methods to quantify microstructures in steel images. In order to accomplish this task, we applied and compared three different CNNs architectures. CNN is a kind of neural network focused on recognizing visual patterns from images. Its training is done without separation between the feature extraction and the classification step, which differs a lot CNN from other image processing algorithms. CNNs are considered state-of-the-art approaches in many vision applications due to their ability to get high accuracy results in complex computer vision problems. The architecture of a CNN is similar to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the Visual Cortex. The basic concept of CNNs was developed by Fukushima (1980). However, only a few years ago CNNs returned to the attention of the scientific community, especially after Krizhevsky et al. (2012), which was a breakthrough in the computer vision area. A typical CNN is compounded by several repeating kinds of layers that are stacked in layers, each named on the basis of the performed operations:

Convolution layer is the first layer to extract features from an input image. It is composed by learnable filters, which are convolved with the input data as shown in Equation(1):

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

where k = 1, ... , K is the index of the k-th feature map in convolution layer and (i, j) is the index of neuron s 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).

Pooling layer progressively reduce the spatial size of the representation, to decrease the number of parameters and computations in the network. The two most common form are max pooling and global average pooling, which either takes maximum or average values in each sub-region of the input data.

Fully-connected Layer is a classic neural network layer, where each node is connected to all nodes of the previous layer. Its output is 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 applies a nonlinear activation over an input signal. It usually follows a pooling or fully connected layer. There are different types of activation functions, such as hyperbolic tangent, and the sigmoid function. However, ReLU function $relu(x) = max(0, x)$ was used because it trains the neural network several times faster without a significant penalty to generalization accuracy.

Dropout Layer is a technique to improve the generalization of CNNs. The method randomly ignores (drop) neurons and their corresponding parameters from the network architecture only during the training phase.

Loss Layer specifies how training penalizes the deviation between the predicted and true class labels and is normally the final layer of a neural network. As our task was to quantify microconstituents in the images, our work can be considered a regression problem. The Mean

Squared Error (MSE) and the Mean Absolute Error (MAE) given by Equation (3) and (4) are two of the loss layers most used in regression tasks.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \tag{4}$$

where n is the number of data points, Y is the true label and $\hat{Y}$ is the predicted label.

In the following subsections, the three CNNs architectures applied in this work are described.

3.1 AlexNet

AlexNet, proposed by Krizhevsky et al. (2012), achieved a top-5 error of 15.3% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), more than 10.8 percentage points lower than that of the runner up. It is considered one of the most influential papers published in computer vision, having spurred many more papers published employing CNNs.

The model is a deep CNN with 60 million parameters and 650.000 neurons. It contains eight layers as shown in Figure 1.

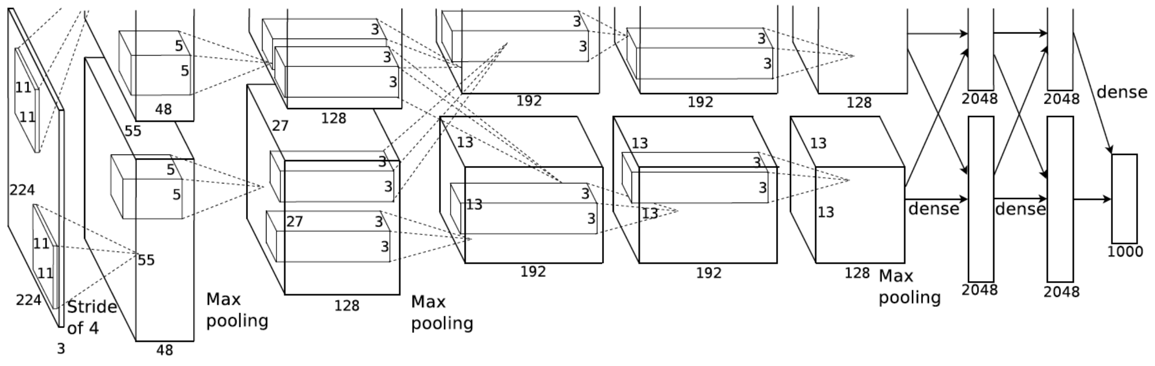

Figure 1: The AlexNet model. (Krizhevsky et al., 2012)

The first five are convolutional layers, some of them followed by max-pooling layers, and the last three are fully connected layers. ReLU nonlinearity is applied after all the convolution and fully connected layers. Dropout is applied before the first and the second fully connected year. It is important to emphasize, as already mentioned, AlexNet was implemented from scratch in this work.

3.2 VGG19

VGG19 is a version of VGG*net*, proposed by Simonyan and Zisserman (2015). It was the runner-up at the ILSVRC 2014 competition, achieving a top-5 error rate of 7.3%. It consists of 16 convolutional, five pooling and three fully-connected layers. The width of the convolution layers starts from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer until it reaches 512. Convolutional layers use 3 x 3 kernels with a stride of 1 and padding of 1 and A rectified linear unit (ReLU) activation is performed right after each convolution. The first two fully-connected layers have 4096 channels each, and the third contains the number of predictable classes, as shown in Figure 2 below.

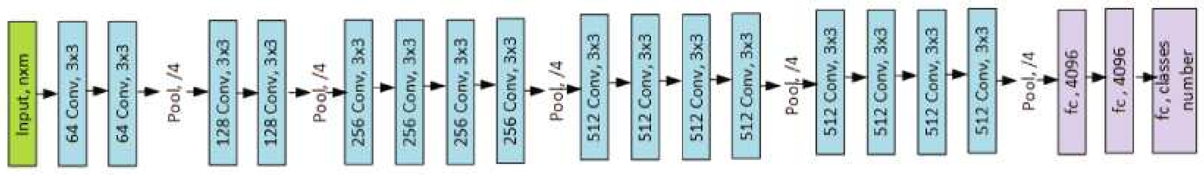

Figure 2: The VGG19 model. (G. et al., 2019)

VGG19 was trained on more than a million images from the ImageNet database. Consequently, the network has learned rich feature representations for a wide range of images. Additionally, as the weight configuration of the VGGNet is publicly available, it was possible to apply the pre-trained version of VGG19 in our task. This process is called transfer learning, which is a machine learning technique based in transferring the parameters of a neural network trained with one dataset for a specific task to another problem with a different dataset and task.

3.3 Xception

Xception, proposed by Chollet (2017) from *Google*, is considered an extreme version of InceptionV3, which is an CNN architecture also by *Google* and it was first runner up in ILSVRC 2015. Xception has 36 convolutional layers structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules, as shown in Figure 3 below.

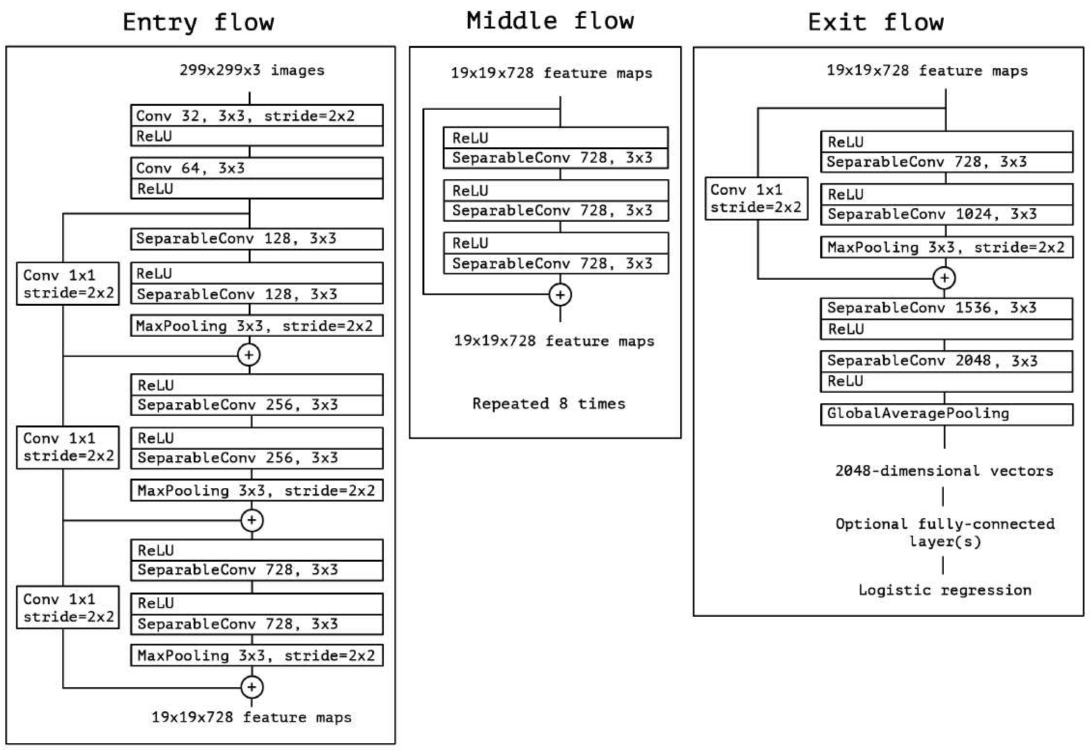

Figure 3: The Xception model. (Chollet, 2017)

Xception was also trained on the ImageNet ILSVRC dataset and it has achieved top-5 accuracy of 0.945, outperforming VGG*net* and InceptionV3. Furthermore, as its weight configuration is also publicly available, we could apply transfer learning again, by using the pre-trained version of Xception in our task.

4 IMPLEMENTATION DETAILS

4.1 Dataset

Our dataset contains 210 micrographs of a 1020 steel welded fusion zones. The images were obtained by an Olympus GX5 Microscope and have a resolution of 2048 x 1532 pixels. Each of these micrographs has a label with the percentage of microconstituents, being so a supervised machine learning problem. These quantities were obtained manually by following ASTM (2012), Figure 4 below shows one of the micrographs of the dataset.

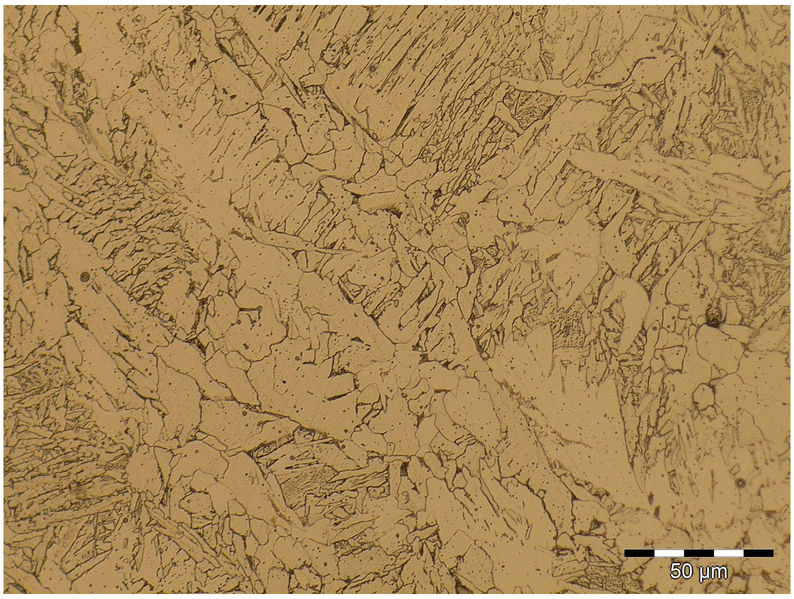

Figure 4: Micrograph image analyzed in this work. Microconstituents: 36% PF(G), 3% PF(I), 16% AF, 34% FS(NA) and 11%FS(A)

4.2 Data Preparation

As CNNs work better with large datasets, augmentation techniques had to be applied to increase the number of images. From the original dataset, 21 images were separated for testing, 21 images for validation, and the remaining 168 images were mirrored in x and y-axis, increasing the data to 672 training images. Furthermore, original resolution of the images was too high and it would have been very computationally expensive to run with these large images. Therefore, we had to choose a lower resolution without compromising results and the best choice was to apply a reduction factor of 0.15, reducing the images to 307x230.

4.3 Training

Python programming language was used to implement the codes and CNN models were built in *Keras* library. In order to train and test CNNs models, NVIDIA Tesla T4 GPU (15079 MiB provided by *Google Colaboratory*) was used. We ran each computational model 100 epochs using 6-fold cross-validation with shuffled data generated by different random seeds. As loss function, MSE was used. Additionaly, MAE and R^2-score (proportion of the variance in the dependent variable that is predictable from the independent variable), were applied as evaluation metrics. Regarding the optimizer, ADAM e Stochastic Gradient Descent (SGD) were applied and compared. In the Adam optimizer, we applied a learning rate of 0.0005 and in SGD we used a learning rate of 0.01, momentum of 0.9, and weight decay of $1 * 10^{-6}$.

5 RESULTS

In Table 1, comparable results between the three CNN models are presented. The results show that AlexNet was the fastest model, taking only 4200 seconds (70 minutes) to run the solution. As expected, Xception network took the longest time as it is the most complex network among the ones we compared. Furthermore, VGG19 with SGD optimizer was the combination, which achieved the best results in terms of error and accuracy.

Table 1: Comparison of results between CNN models

Architecture	**Optimizer**	**Execution Time (s)**	**Average R2 score**	**MAE (%)**
VGG 19	ADAM	8399	0.796	5.74
Xception	ADAM	10798	0.769	5.57
AlexNet	ADAM	**4200**	0.793	5.53
VGG 19	SGD	8404	**0.834**	**4.88**
Xception	SGD	10806	0.7352	6.36
AlexNet	SGD	4205	0.789	5.81

In order to show how the model can predict the percentage of microconstituents very accurately, we present predictions of VGG19 model for six images of test, Figures 5 to 10. Each micrograph is presented with its respective percentage label, the model predictions and the mean absolute error. The quantities are presented in the following form: [%PF(G), %PF(I), %AF, %FS(NA), %FS(A)].

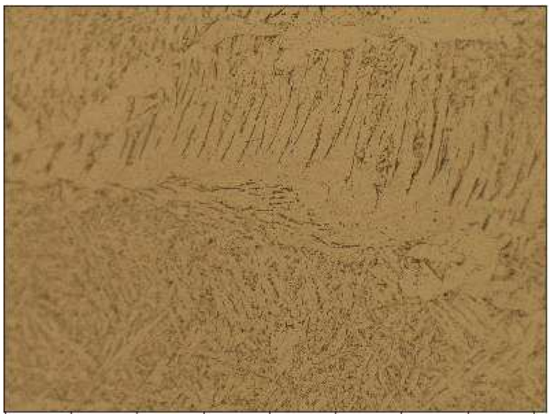
Figure 5: Manual quantification:
[49%, 5.7%, 12%, 29.2%, 4.2%]
Predicted quantification:
[52.6%, 5.4%, 12.7%, 25.1%, 4.2%]
MAE: 1,76%

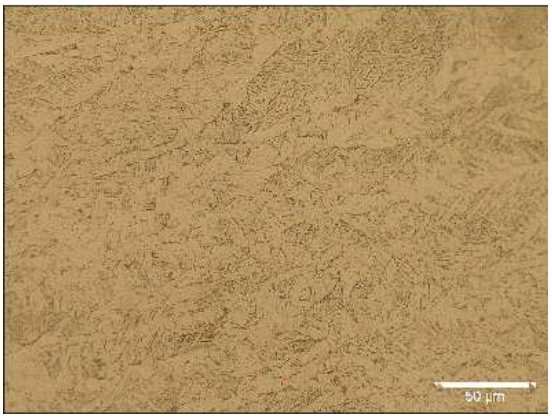

Figure 6: Manual quantification:
[31.2%, 4.2%, 7.8%, 47.4%, 9.4%]
Predicted quantification:
[32.9%, 5.4%, 13.5%, 42.1%, 6.1%]
MAE: 3,43%

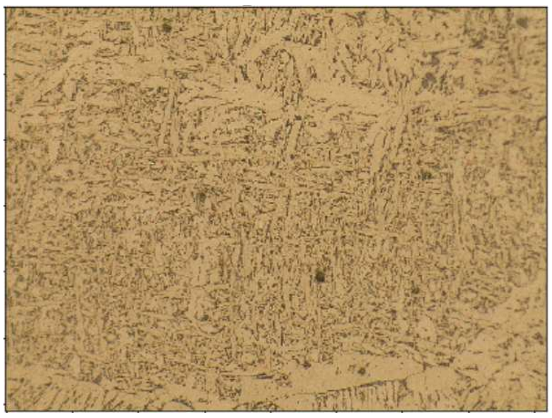
Figure 7: Manual quantification:
[22.4%, 3.1%, 19.3%, 45.8%, 9.4%]
Predicted quantification:
[25.8%, 4.2%, 18.2%, 46.9%, 4.9%]
MAE: 2,25%

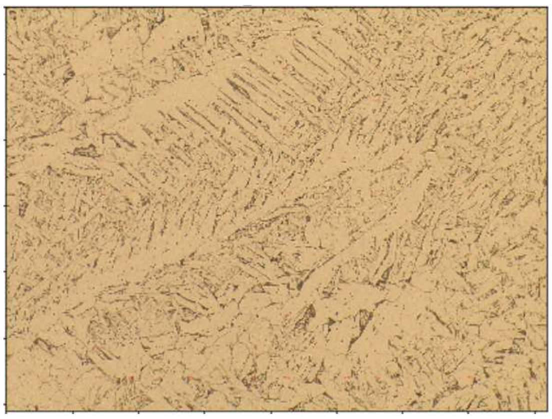
Figure 8: Manual quantification:
[34.4%, 4.2%, 12%, 39.1%, 10.4%]
Predicted quantification:
[34.1%, 5.5%, 13.1%, 41.2%, 6.2%]
MAE: 1,81%

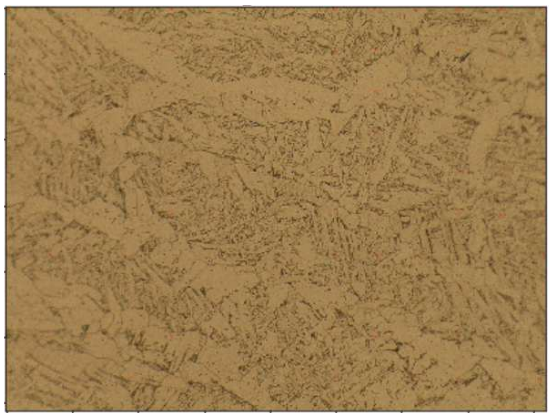
Figure 9: Manual quantification:
[47.4%, 7.8%, 11.5%, 32.3%, 1%]
Predicted quantification:
[43.9%, 7.3%, 11.9%, 32.9%, 4.2%]
MAE: 1,63%

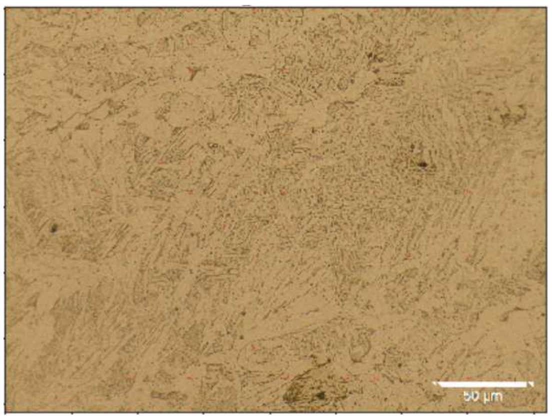

Figure 10: Manual quantification:
[31.5%, 3.7%, 13.9%, 41.7%, 9.3%]
Predicted quantification:
[29.6%, 7%, 14.7%, 42.8%, 5.9%]
MAE: 2,11%

By analyzing the results, it is possible to observe that the model can reach values very close to the input value. However, it is also observable that the CNN model mistakes more in some microconstituents than others. Our models better predict the microconstituents that have less variability among the data, such as PF(I) and FS(A). In contrast, microconstituents such as FS(NA) and PF(G) present large variability among the data, which makes the task more difficult.

To illustrate this scenario, Figure 11 presents a boxplot of the average error of each model per different microconstituents. As expected, it can be seen that models better predict PF(I) and FS(A), and mistake more in PF(G) and FS(NA), as mentioned above.

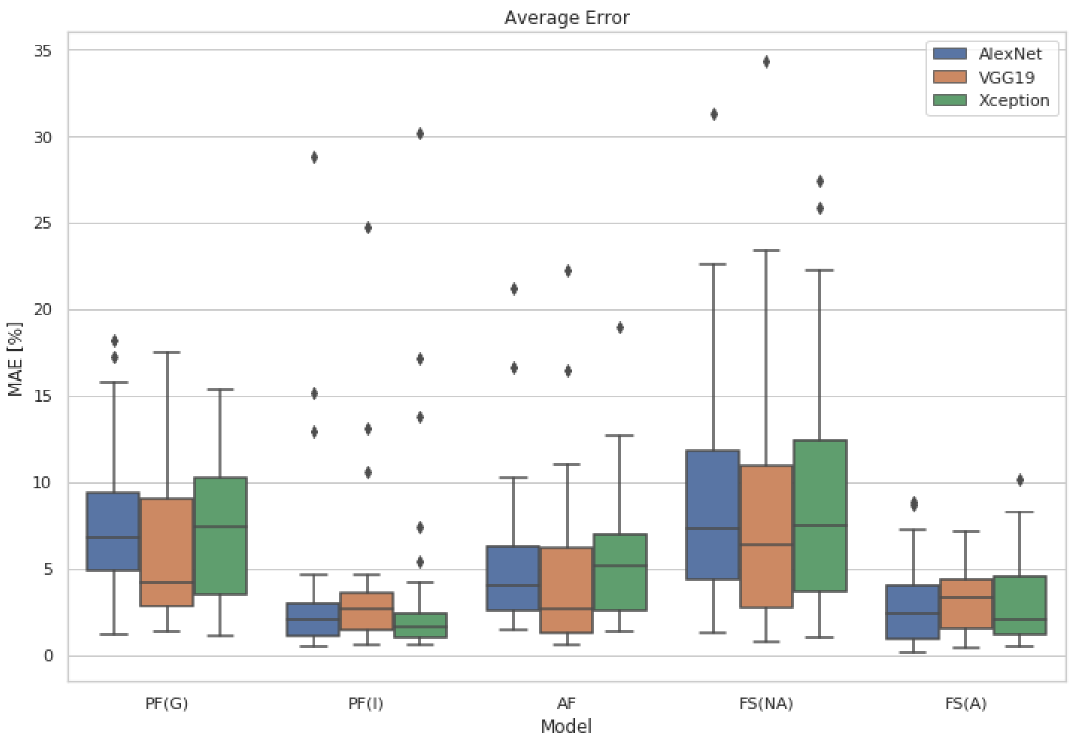

Figure 11: Average error per microconstituent

6 CONCLUSIONS

This work presents the feasibility of applying CNNs for steel microstructural quantification. In order to achieve this goal, we have compared the performance of three CNN models. In addition, augmentation techniques were used to increase the size of the training set and consequently improve the performance of the models. According to results, VGG19 was considered the best model for this application. Additionally, it was shown that the performance of the models varies for each micro constituent, depending on their variability among the data.

From the predictions in the test set, it was observed that the model can achieve output values very close to the input values, proving the efficacy in applying deep learning methods in this task. In addition, it is worth mentioning that the network performs the quantification automatically in a few seconds, and this task is currently done manually and consequently, very time consuming and prone to error.

REFERENCES

ASTM. *Standard Test Method for Determining Volume Fraction by Systematic Manual Point Count*. American Society for Testing and Materials, 2012.

Azimi S.M., Britz D., Engstler M., Fritz M., and Mücklich F. Advanced steel microstructural classification by deep learning methods. *Scientific Reports*, 8:2128, 2018.

Bhadeshia H. and Honeycombe R. *Steels: Microstructure and Properties*, volume 4. Elsevier - Butterworth-Heinemann, 2017.

Campbell A., Murray P., Yakushina E., Marshall S., and Ion W. New methods for automatic quantification of microstructural features using digital image processing. *Materials and Design*, 141:395–406, 2018.

Chollet F. Xception: Deep learning with depthwise separable convolutions. *IEEE Conference* *on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chowdhury A., Kautz E., Yener B., and Lewis D. Image driven machine learning methods for microstructure recognition. *Computational Materials Science*, 123:176–187, 2016.

DeCost B.L., Francis T., and Holm E.A. Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. *Acta Materialia*, 133:30–40, 2017.

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

G. L., G. R., P. S., C. P., B. F., and M. F. Automatic microstructural classification with convolutional neural network. *Information and Communication Technologies of Ecuador (TIC.EC)*, 884:170–181, 2019.

Gola J., Britz D., Staudt T., Winter M., and Andreas Simon Schneider Marc Ludovici F.M. Advanced microstructure classification by data mining methods. *Computational Materials Science*, 148:324–335, 2018.

Krizhevsky A., Sutskever I., and Hinton G.E. Imagenet classification with deep convolutional neural networks. *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1:1097–1105, 2012.

Simonyan K. and Zisserman A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations ICLR*, 2015.