MSc-Project: “Image Segmentation of the γ’-Phase in Nickel-base Superalloys utilizing Deep Learning”

Motivation

Mechanical properties of a material are significantly determined by its microstructure. Thus, microstructural analysis as a technique is widely spread and well known in practice. This analysis is mostly used as a tool of quality assurance and thus is directly responsible for the safety and scrap rate of products. Ever since then it has to be carried out by human experts in a manual fashion. Not only does it take a huge amount of time and money to train and to pay personnel for this task, its performance may also vary from time to time and is in general rather subjective. At first it may seem to be obvious for companies to somehow automate this process yet only little work in this field has been done. This is mainly caused by the complexity of the possible ways the γ'-phase and adjacent substructures represents itself in the material. Therefore, the following research questions the thesis tries to answer:

Is it possible to apply an image segmentation technique based on Deep Learning to determine different phenotypical structures of a material with nearly human-like precision?

An automatic solution would produce a faster, cheaper and most importantly a more consistent segmentation map. This approach is cost efficient as it does not rely on any domain knowledge or human

Problem description

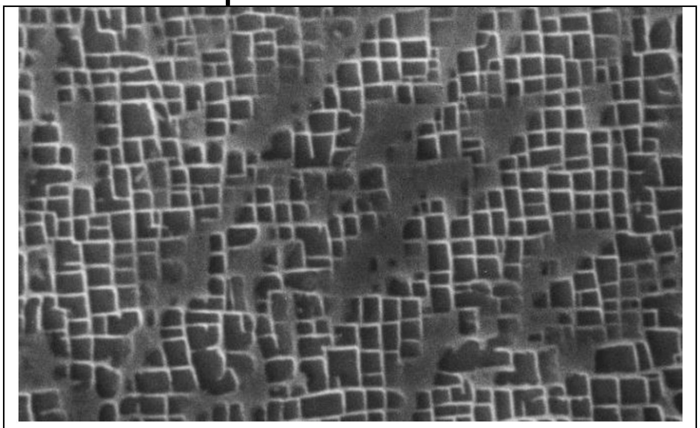

Figure 1: Nickel-base superalloy microstructure showing cubic γ’-phase and embedding matrix

Nickel-base superalloys can be precipitation hardened by the so called gamma‘ phase, which manifests itself in cubic precipitations. Unfortunately, in metallographic images not all four edges are always continuously detectable or even visible, making an automatic solution quite challenging.

Furthermore gamma' precipitates can occur on two distinct size scales, with a need to distinguish both. Thirdly the gamma' precipitate can deform with temperature and applied force, the so-called rafting.

Metallographic images occur in different magnifications and only show part of the whole prepared surface, leading to possibly very different phenotypic occurrence of the microstructure

All those effects have to be accounted in the metallographic analysis and a robust way to identify gamma' precipitates and distinguish this from the surrounding matrix needs to be found.

Time schedule

The following tasks are foreseen:

1. General introduction into TensorFlow (https://www.tensorflow.org/guide/low_level_intro)
2. Use Estimators to train a simple CNN on the MNIST-Dataset (https://www.tensorflow.org/tutorials/estimators/cnn)
3. Training of an advanced deep-CNN on the CIFAR-10 Dataset, which also talks about TensorBoard (https://www.tensorflow.org/tutorials/images/deep_cnn)
4. Have a look at the DeepLab implementations in the research models of TensorFlow to get an understanding of how transposed-convolutions (also sometimes called upconvolutions) work (https://github.com/tensorflow/models/tree/master/research/deeplab)

The student will first familiarize with the methodology to analyze gamma' metallography images. From this the first task is to use the existing image analysis infrastructure available at MTU to build a working demo code to segment images, based on the already labeled images at MTU. For this a reasonable network architecture needs to be chosen and implemented. The influence of the tunable parameters, like number of layers, stride, size of the convolutions or amounts of max-pooling on the accuracy and run-time shall be investigated.

As Convolutional Neural Networks have a predetermined intake of pixels different strategies to handle different images sizes, shall be investigated and compared regarding accuracy, runtime, memory usage and ease of implementation in a professional environment.

Upon completion of the foremost tasks, the implementation of other Network architectures, with diverging segmentation strategies can be looked at. Additionally, the code be expanded to not only a continuous segmentation, but an instance-based segmentation like Mask-R CNN.