DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2017

Automated Classification of Steel Samples

An investigation using Convolutional Neural Networks

BJÖRN T.I. AHLIN

E. MARCUS GÄRDIN

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Abstract

Automated image recognition software has earlier been used for various analyses in the steel making industry. In this study, the possibility to apply such software to classify Scanning Electron Microscope (SEM) images of two steel samples was investigated. The two steel samples were of the same steel grade but with the difference that they had been treated with calcium for a different length of time.

To enable automated image recognition, a Convolutional Neural Network (CNN) was built. The construction of the software was performed with open source code provided by Keras Documentation, thus ensuring an easily reproducible program. The network was trained, validated and tested, first for non-binarized images and then with binarized images. Binarized images were used to ensure that the network's prediction only considers the inclusion information and not the substrate.

The non-binarized images gave a classification accuracy of 99.99 %. For the binarized images, the classification accuracy obtained was 67.9%.
The results show that it is possible to classify steel samples using CNNs. One interesting aspect of the success in classifying steel samples is that further studies on CNNs could enable automated classification of inclusions.

KEYWORDS: CNN, Automated classification, Inclusions, Steel.

Sammanfattning

Automatiserad bildigenkänning har tidigare använts inom ståltillverkning för olika sorters analyser. Den här studiens syfte är att undersöka om bildigenkänningsprogram applicerat på Svepelektronmikroskopi (SEM) bilder kan klassificera två stålprover. Stålproven var av samma sort, med skillnaden att de behandlats med kalcium olika lång tid.

För att möjliggöra den automatiserade bildigenkänningen byggdes ett Convolutional Neural Network (CNN). Nätverket byggdes med hjälp av öppen kod från Keras Documentation. Detta för att programmet enkelt skall kunna reproduceras. Nätverket tränades, validerades och testades, först för vanliga bilder och sedan för binariserade bilder. Binariserade bilder användes för att tvinga programmet att bara klassificera med avseende på inneslutningar och inte på grundmatrisen.

Resultaten på klassificeringen för vanliga bilder gav en träffsäkerhet på 99.99%. För binariserade bilder blev träffsäkerheten för klassificeringen 67.9%.
Resultaten visar att det är möjligt att använda CNNs för att klassificera stålprover. En intressant möjlighet som vidare studier på CNNs kan leda till är att automatisk klassificering av inneslutningar kan möjliggöras.

Nyckelord: CNN, Automatisk klassificering, Inneslutningar, Stål

List of Abbreviations

CNN	Convolutional Neural Network
AI	Artificial intelligence
ML	Machine Learning
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
FC	Fully Connected
ReLU	Rectified Linear Unit
SEM	Scanning Electron Microscope
SE	Secondary Electrons
BSE	Back-Scattered Electrons
API	Application Programming Interface

Table of Contents

1. Introduction

Non-metallic inclusions in steel can significantly affect the properties regarding mechanical- and fatigue behaviour of the steel. Increasing requirements on material properties demand that classification of inclusions with regards to size and characteristics needs to be performed [1]. This classification is often done manually, which can be tedious and time-consuming. If this process could be performed automatically by a computer it would radically decrease the time spent on classification.

It is the subfield of Artificial Intelligence (AI) called Machine Learning (ML) that provides the possibility to automate classification of inclusions. Earlier studies have confirmed that the ML archetype Artificial Neural Network (ANN) can be applied as a modelling tool for prediction of temperature in different processes in the steelmaking industry [2] [3] [4]. For image recognition, ANNs can be applied. Some examples of previous studies with ANNs has concerned the detection of rust defects on steel bridge coatings [5], to detect internal defects in castings with X-ray images [6], Classification of corrosion defects in NiAl bronze [7] and the inspection of surface damage on engineering ceramics due to grinding [8]. Recent studies have shown that for image recognition the ML archetype Convolutional Neural Network (CNN) provides better predictions than other neural networks [9].

Automated image recognition with CNNs has previously been successfully applied in the steel industry. Some studies that have been conducted using the CNN architecture includes microstructure recognition [10], to detect and classify surface defects on the surface of steel strips [11] [12], classification of surface defects on hot-rolled steel sheets [13] and localizing slab identification numbers in industrial settings [14]. The belief is that the usage of machine learning models will increase and become a valuable resource for the steelmaking industries. The present project will function as a pilot study to investigate if CNNs can be applied to classify inclusions.

The project's main purpose is to build an automated image recognition program with CNN topology. With the goal to enable classification of two steel samples with high accuracy. The program will be created using open source code, thus making it easily reproducible.

This thesis will also consist of a study on different types of inclusions. Investigation of what causes inclusions, the morphology of different types of inclusions and how inclusions affect the properties of steel will be done. If the model is successful in classifying two steel samples, it could give a basis for further work on more advanced deep learning models that possibly can classify inclusions based only on image analysis, thus reducing the need for conventional chemical analysis.

The report will also give a short background into the field of machine learning, what it is and what it can do. The background will provide readers with no previous experience in the field of machine learning some understanding. The study will hopefully raise an interest in what is thought of as the future of computer science technology applied to industrial problems.

2. Background

2.1 Artificial Intelligence

Artificial Intelligence (AI) has intrigued people ever since the concept of computers was conceived. Today AI is a flourishing field with various possible applications. Some examples are understanding speech or images, accommodate in scientific research and to automate conventional labour.

AI has from the beginning been good at solving problems that can be described by formal mathematical rules. The real challenge for AI is to tackle problems that are difficult to describe formally, which are problems that humans solves intuitively, such as recognizing speech or details in images. A solution to these problems is to let computers learn from experience and adapt to new circumstances. This can be facilitated by the statistical approach to AI which is called Machine Learning [15].

2.2 Machine Learning

Machine Learning (ML) concerns the creation of software that automatically improves when introduced to new examples, i.e. data points. The goal is to get the program to learn and thus give it the ability to adapt to new circumstances. This is achieved by analyzing data to find important patterns and trends. There are numerous applications for which ML is suitable. Examples include, but are not limited to, computer vision, speech recognition, signal processing, robotics, industrial optimization, fraud detection and finance [2].

ML can roughly be split up into three categories: Supervised learning, unsupervised learning and reinforcement learning. The names of the categories indicate the different methods of learning.

- *Supervised learning* knows the correct output data point for each input data point during training. The goal is to make the model generalize. The generalizing is to accurately map new input data points to its correct output values. The model's performance during the learning process is often evaluated by dividing the data set into three categories: Training set, validation set and test set. In general, this dividing of the data set is done in order to fit the model with the training set, i.e. to train and adjust the weights of the network. Then estimate prediction errors during training of the model using the validation set. The test set is then introduced to the model to assess the true generalization error and is the final test to see if the model is performing well on previously unseen data [16].

- *Unsupervised learning* models do not, in comparison with supervised learning, know the specific output values to respective input data point. These types of models can be

applied to reveal patterns and underlying structures in the data. Some examples of applications are breast cancer research and search patterns in search engines [17].

- In *reinforcement learning*, the program does not know the correct output at all and is forced to find the optimal output by trial-and-error. The model is built so that it receives rewards when providing a solution that gives wanted results. If the model gives results that are not wanted it will be penalized [18].

In this study, the category supervised learning will be used. The training of any supervised machine learning model requires a substantial amount of data. The amount of data is one of the major influences on how well a model will perform. More data leads to better training and therefore better predictions.

Supervised ML models can be used for regression and classification problems. Regression problems are when the wanted output consists of one or more continuous parameters. An example is temperature prediction in steel processes. Classification problems are when the desired output is assigned to a finite number of categories such as object recognition in images [18]. There are numerous ML models which can be applied to the different problems, but the scope of this thesis will not cover all of them. In this study, one model that can handle both regression and classification called Artificial Neural Networks (ANN) and some of its variants will be further discussed in the following section.

2.3 Artificial Neural Networks

ANN are structured with the biological cell communication as an influence. It is interconnected neurons that enable this communication. ANNs mimics this evolutionary trait by connecting input nodes to output nodes, in which each node represents a neuron. The sending of signals from neuron to neuron in ANNs is facilitated by the use of a transfer function. There are several different types of transfer functions that can be used. One of the most common transfer function is the logistic sigmoid function which is given by the following equation:

$$\text{output} = \frac{1}{1 + e^{-\left(\sum_i w_i x_i + w_0\right)}} \tag{1}$$

Where w_i is the weight for each input, x_i and the bias weight w_0.

Some of the inputs have greater importance to the output, called an activation, than others. This is modelled by weighting the input of each neuron. The weights of the inputs are summed up and the chosen transfer function sends the output signal to the next neuron [19]. The activation output is given by equation 2:

$$f\left(\sum_{i=o}^{3} w_i x_i\right) = \text{Output} \tag{2}$$

A simple representation of a network with one node and activation output from equation 2 is shown in figure 1.

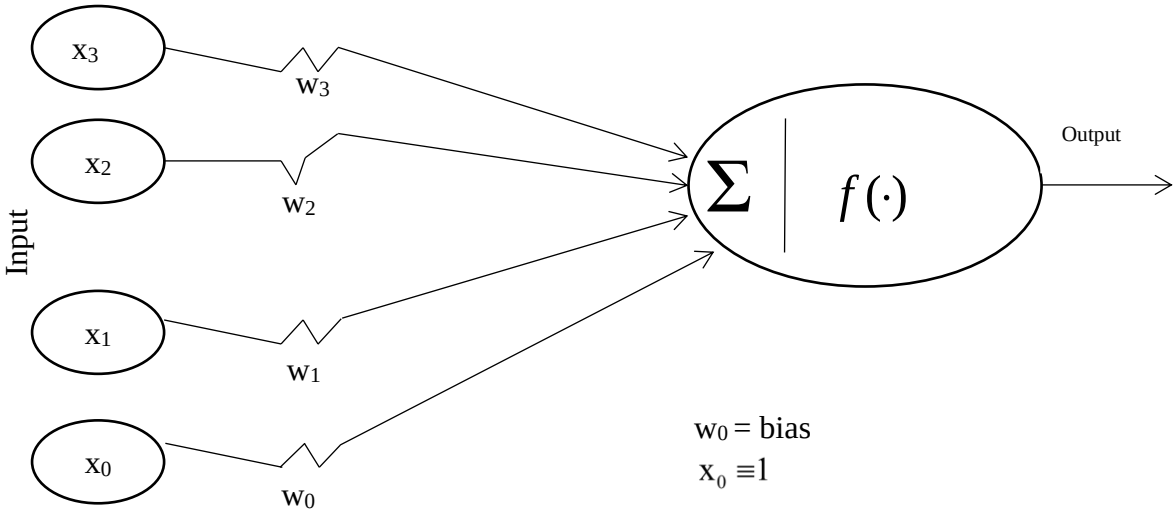


Figure 1: Neuron model with weighted inputs and embedded transfer function. The oval shapes each represents a single neuron.

ANNs are constructed using multiple layers, divided into three groups: Input layer, hidden layers and output layer. The input layer distributes the input data to the first hidden layer. The number of neurons in the input layer is equivalent to the dimension (number of variables) of the input data. The bulk of layers in an ANN are the hidden layers. These layers process the input. The output is then sent forward to another hidden layer to function as input, where the same procedure is performed. The output from the last hidden layer is sent as input to the output layer. This is called a feedforward neural network. The data is transferred in only one direction. Travelling from input layer through hidden layers forward to the output layer without any loops or cycles. In figure 2 a general topology of an ANN is depicted.

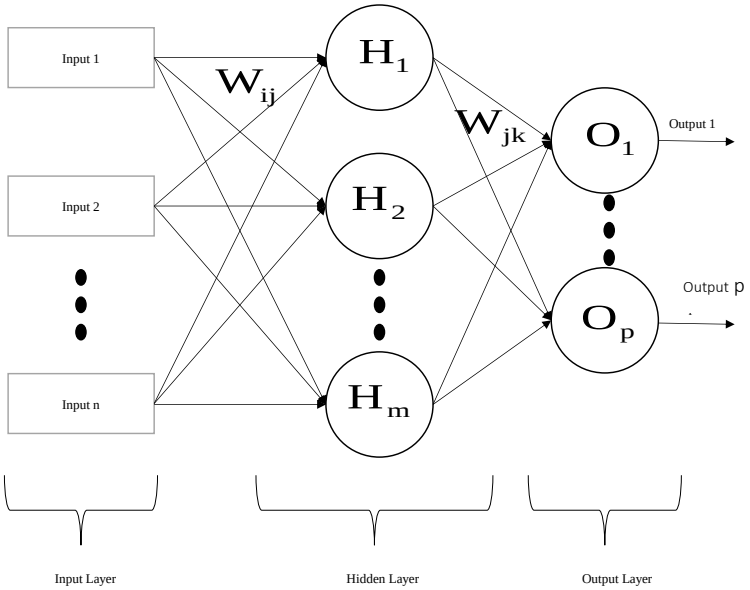


Figure 2: A general topology of a feedforward ANN with n input neurons, m hidden neurons and k output neurons.

The topology in figure 2 is one example of all the possible ways to structure the network. The programmer can, for example, specify the number of hidden layers, the number of neurons in the layers and the number of output categories. The connections between neurons can be changed in order to send signals back through the network and between neurons in the same layer. These types of networks are called Recurrent Neural Networks (RNN) [19].

One topology often used is the Fully Connected Network. In such a model, all the neurons in adjoining layers are connected. Since all the neurons are connected between layers, the number of parameters increases exponentially with every added layer. A simple calculation for a three-layer ANN, with hundred neurons in each layer, shows this substantial increase in parameters: $100 \cdot 100 \cdot 100 = 10^6$ parameters. The huge amount of parameters leads to a great computational cost and can also contribute to problems with overfitting [17].

As mentioned earlier, recent studies have shown that the ANN topology Convolutional Neural Network (CNN) has greatly improved the accuracy of image classification. CNN requires just a fraction of the amount of parameters in a corresponding fully connected ANN [15]. CNNs will be thoroughly discussed in the following section.

2.4 Convolutional Neural Networks

Convolutional neural networks are networks specialized for handling information that has a grid-like topology. As the name of the network implies, a linear operation named convolution is applied to the data.

CNNs architecture is composed of four different kinds of layers. The layers are called input, convolutional, pooling and fully connected (also known as dense). It is the convolutional, pooling and fully connected layers, except output, that corresponds to the hidden layers in a CNN. The before mentioned layers will be further explained in the following subsections. In figure 3, a general structure of a CNN is displayed [20].

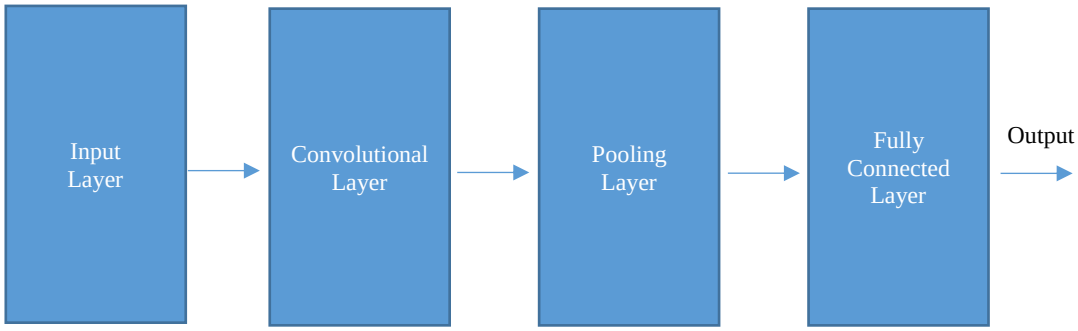


Figure 3: A general structure of a CNN with four layers.

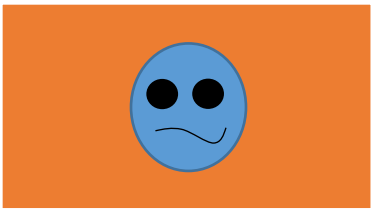
The programmer decides how many and what kind of layers that are needed to tackle the problem at hand. The pattern in the layered architecture can also differ between networks. It is possible to put multiple convolutional layers before a pooling layer. Multiple convolutional

layers are well suited for larger and deeper networks when more detailed features need to be extracted from an image [21].

A previous belief was that more layers provide better predictions. Recent studies have shown that just piling layer upon layer does not improve accuracy. The study by He, K et al. show that around 150 layers are the limit for accuracy improvement on CNNs [22].

2.4.1 Input Layer

The input layer is where the data points serving as input is introduced to the model. In this study, the input data is images. The images will for the computer be seen as an array of pixel values. In figure 4 a representation of an image as humans visualize it and what the computer "visualize".



What humans visualize

49 64 93 56 05 04 75 15
81 04 85 15 86 10 45 61
78 56 78 16 57 66 67 56
10 73 56 80 37 08 71 08
73 05 75 31 87 58 54 81

What computers "visualize"

Figure 4: Representation of how an image is seen by computers compared to how humans visualize it.

2.4.2 Convolutional Layer

The purpose of the convolutional layer is that it applies a set amount of convolutional filters on the image and performs linear computations to the input array. The number of filters is up to the programmer to specify. The filters extract features from the image and create a corresponding feature map to each filter [20]. The extracted features correspond to the largest changes in the images, which are gradients between different regions in the image. For example, if there is an image with white background and a straight black line in the middle, the largest gradient will be found at the interface between them. The convoluted data, i.e. the linear feature maps, is then processed by a transfer function called Rectified Linear Unit (ReLU). ReLU introduces non-linearity to the network since most of the data that CNNs process is non-linear. It corrects the feature maps with a threshold operation, any input value less than zero is set to zero thus ensuring that the feature maps are always positive [21]. Figure 5 shows the effect of the ReLU function.

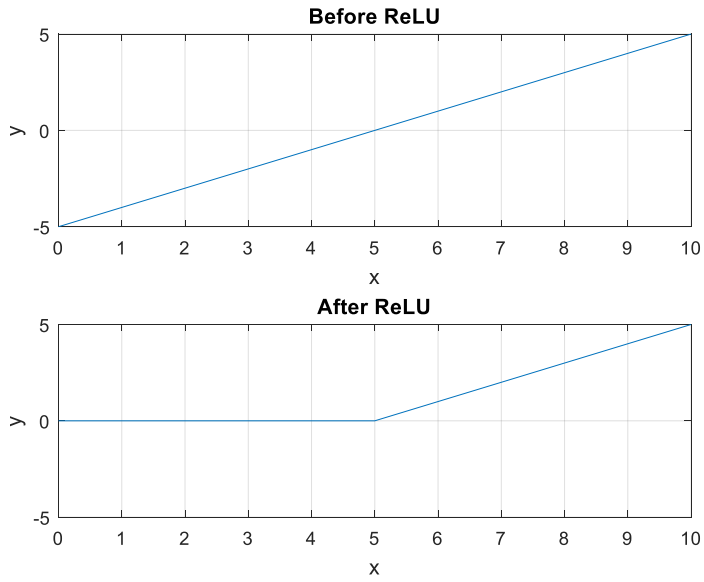


Figure 5: How the ReLU function applies to input data.

The feature maps, produced by the convolving and ReLU, are the input to the pooling layer. One way to visualize a convolutional layer is to think of the filter as a microscope that amplifies a specific region, called the receptive field, on the input data. The array of the image has the dimensions W x H x D, where W is the width of the image, H is the height of the image and D is the number of colour channels. The filter is a matrix of numbers, called weights. The filter is smaller in spatial size than the input matrix but must have the same number of dimension. The filter will move over the image and perform elementwise multiplications between the values in the filter with the original pixel values. The number obtained correspond to a high activation value in the original input. How the filters move over the input array is specified by what is called strides. Each stride moves the filter a specified length, often 2 pixels. The filter focuses on the specific region reached and calculates an activation value for each region. The specific numbers calculated create a new array of output, a feature map, which is smaller in size than the input array [20]. A visualization of the convolutional layer, with a 3x3 filter and stride of 1 on a 5x5 input array, where the filter reduces the array from 5x5 to 3x3, with the filters first two strides are depicted in figure 6.

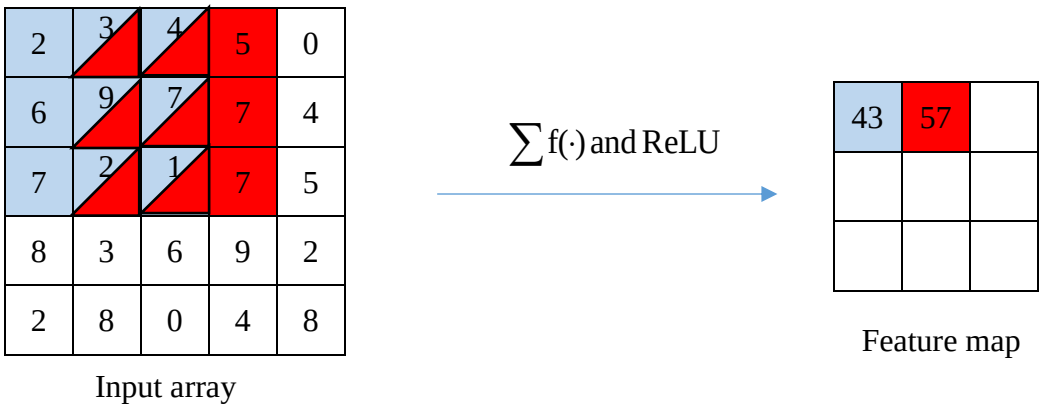


Figure 6: Visualisation of a convolutional layer with a 3x3 filter and a stride of 1. Blue represents the first stride of the filter and the red represent the second stride.

The output obtained from the convolutional layer is then sent to the pooling layer, or another convolutional layer, for further processing.

2.4.3 Pooling Layer

The pooling layer purpose is to downsample the output from the feature maps. The idea is to apply a filter, usually with the size 2 x 2 and a stride with the same length, 2, on the output from the convolutional layer. By using a max pooling function, the highest value from the feature map is extracted. The idea is that once the specific feature from the original input is known, the relative location towards other features is more important than its exact location. Pooling significantly reduces the dimensions of the input volume [21]. This is shown in figure 7.

Figure 7: Pooling example by a 2x2 filter with stride of 2.

Pooling reduces the parameters by 75% and ideally obtains the most important features. The reduction decreases the computational cost of the model. One other aspect of pooling is that it decreases the problem of overfitting. Regarding overfitting, think of it as a simple curve fitting. A 7^{th} power curve fits exactly to eight data points. When the system is introduced to new data the curve will have a problem adjusting. A visualization of this is shown in figure 8.

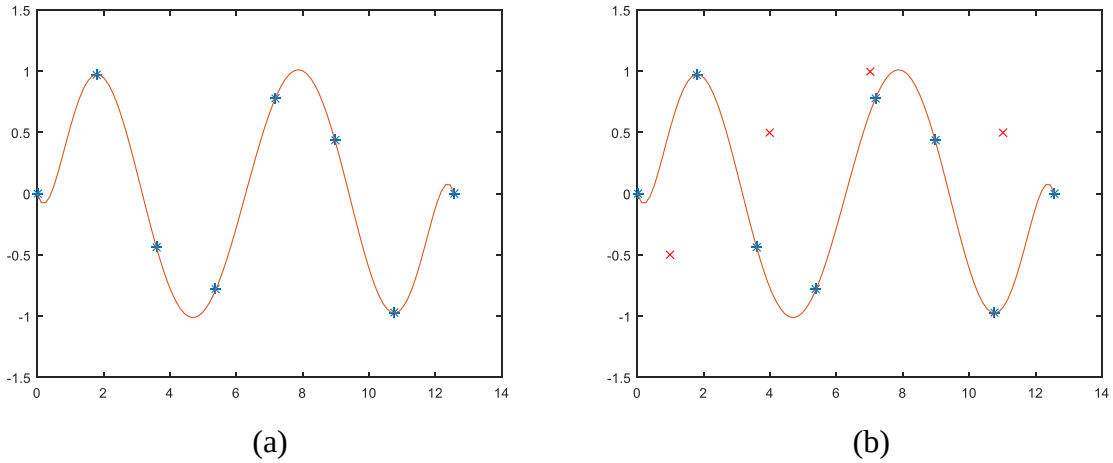


Figure 8: A representation of overfitting. (a) 7^{th} power curve fitted to 8 data points. (b) The same fitting with additional data points.

In machine learning, overfitting can lead to a very good generalization on the training set. However, when it comes to validation and testing the model does not generalize properly [20].

One issue with pooling can be that some information from the original input is lost. But the reduced risk of overfitting is often preferable compared to the possible reduction of important information.

2.4.4 Fully Connected layer

The name Fully Connected (FC) implies that every node in the previous layer is connected to every neuron in the current layer [23]. To be able to connect every node on the first FC layer to the preceding layer, the outputs multidimensional arrays must be put in a single array. This is accomplished by applying vectorization to the matrices to perform a linear transformation into a single row vector. A visualization of flattening is displayed in figure 9.

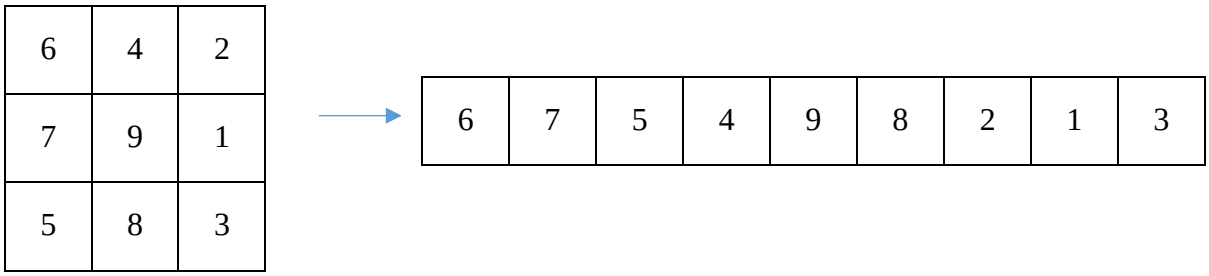


Figure 9: Vectorization of a 3x3 matrix into a row vector.

Every neuron in the FC layer uses all the output from the previous layer as input. All the connections lead to a great increase in parameters for the network to process. The input is weighted and an activation function, normally ReLU, is applied to determine the output. The first FC layer is often followed by additional FC layers, which have fewer neurons. This is to extract the highest activation output and to decrease the amount of parameters.

2.4.5 Output Layer

The output layer, or classification layer, is also an FC layer. It sorts the output from the final hidden layer into different categories using a different activation function than in the previous layers of the network. The activation function that is commonly used in this step is the softmax function. This function compresses the output from the last hidden layer into probability values in the different categories. Therefore, the sum of the outputs is always equal to 1 [24]. The output from the classification layer corresponds to the class with the highest probability.

2.5 Training a CNN with Supervised Learning

The training of a CNN using supervised learning is, just like traditional ANN, an iterative process. The training is conducted with labelled data, meaning that the network knows what output it should give a certain input. Training is divided into epochs where all the training data is used in every epoch. With each iteration, the program changes the weights of the network according to the inequality between its predicted output compared to the labelled output. The weights will continuously be corrected until the calculated output is satisfactorily close to the desired output or until all epochs are finished [23].

The weights need to be set before training, otherwise, the training does not have anything to adjust. The weights are set to random values. These values must be small in order to prevent saturation. Saturation means that a neuron is highly active or inactive for all input patterns and thus impossible to train. Too many saturated neurons will make the whole network impossible generalize [23].

The actual training is facilitated by what is called backpropagation algorithms. Backpropagation consists of four different steps:

1. The first step is the forward pass, or forward propagation, where the training input is fed through the whole network and an output is obtained.

2. The second step is the loss function, also called error function. The loss is obtained by comparing the predicted output with the labelled output. It is often defined by a mean squared error function. The loss will be huge for the first couple of iterations. The goal is to minimize loss and get a prediction output as close as possible to the labelled output.

3. Minimizing loss is the same as an optimization problem in calculus, with the goal to find which weights contributes most to the error of the network. The problem can be described by equation 3.

$$\frac{dL}{dW} = \text{constant} \tag{3}$$

   Where L and W are loss and weights, respectively. For every iteration, this derivative is updated, with the goal to reach a constant as close to zero as possible. This means that the optimizer searches for a global minima in loss space. The third step consists of backward pass, or backpropagation, where the loss value is propagated backwards in the network. At each node, an error value for every neuron is calculated based on the contribution to the total loss, hence obtaining a partial derivative for loss over weight for specific neurons.

4. The partial derivative obtained in the third step is then used in the fourth and final step, which is the weight update. This is where all the weights in the network are corrected. The weight update is described by equation 4.

$$w = w_i - \eta \frac{\partial l_i}{\partial w_i} \tag{4}$$

Where w is the corrected weight, w_i the initial weight, l_i is the loss for specific neuron i and η is the learning rate.

The learning rate is specified by the one constructing the network. A high learning rate means that greater steps are taken in the correction of weight, which makes the model converge faster than with a low learning rate. The problem that can occur with a high learning rate is that the corrections are too large and not precise enough to obtain satisfactory results.
The network will adjust its weights for the specified number of iterations. After the weights update the system needs to be validated. The validation takes place at the end of every epoch. It is performed with labelled data to evaluate how well the model generalize after each epoch. The network's training is completed once all the epoch has been computed [18].

After the training, a labelled test batch is introduced. However, no training is conducted with the input. The predicted output is then compared to the labelled output and prediction accuracy is obtained.

3 Inclusions

This section will discuss inclusions, how they are formed, how they influence the mechanical properties of steel, different morphologies of inclusions, and the effect of calcium treatment on inclusions.

3.1 Categorization and formation

Inclusions are in general classified into two main groups, endogenous and exogenous.

Endogenous inclusions are the products of chemical reactions in the melt. These particles are relatively small and are in general dispersed throughout the steels. Endogenous inclusions are not counted as defects on a macroscopic level but have an impact on the characteristics of steel.

In iron smelts, the solubility of oxygen is about 0.16% and increases with temperature. When the iron solidifies, the solubility of oxygen is much lower, almost zero. If the liquid iron starts to solidify with a high amount of oxygen dissolved, a high part of the oxygen tends to precipitate as spherical FeO-inclusions in the solidified iron [25]. In steel smelts, the oxygen content in liquid steel is lower than in pure iron, about 0.01-0.1% depending on the steel's composition. During the solidifying of steel, some parts of the oxygen will be dissolved and the parts not dissolved may be present as small oxide inclusions.

A common way to lower the oxygen content in the steel is by the deoxidation process. The process is made by adding elements with a higher affinity than iron. There are two types of deoxidation processes. One is called primary deoxidation, which is a process that occurs when elements, such as calcium, carbon and aluminium are introduced to the smelt. The deoxidation elements create inclusions with a lower density than the liquid steel. The density difference makes the inclusions rise in the smelt towards the metal-slag interface. Secondary deoxidation happens when the oxygen solvability decreases during solidification [25].

Exogenous inclusions are entrapment of non-metallic substances, such as slag and dross residues, but could also be parts of the molding material. The inclusions vary greatly in size and are often in clusters. The size and clustering make them easy to locate in the process of steelmaking [26].

3.2 Effect on mechanical properties

Inclusions in a material behave as stress raisers and tend to affect mechanical properties negatively. Due to the incompatibilities with the steel matrix, inclusions create stress fields around itself that are potential crack initiation sites. Examinations of stress fields are usually made by examining the morphology of the inclusions, the inclusions mechanical behaviour, and the interference between inclusions and the matrix [27]. Inclusions with different

composition affect the mechanical properties of steels differently. For example, manganese sulphides (MnS) tends to elongate when machined, having a high influence on local ductility [28]. Alumina (Al_2O_3), on the other hand, are hard, small and tends to cluster, having a negative effect on fatigue strength [29].

3.3 Morphology of inclusions

Inclusions may have irregular shapes and therefore a 2D analysis is not preferable when classifying morphology. Electrical extraction onto a film filter is a 3D technique often used to see more of the morphology and get a broader picture of the stress field [30]. However, 3D analysis techniques take longer time than 2D analysis techniques. In figure 10 the difference between 2D and 3D images is depicted.

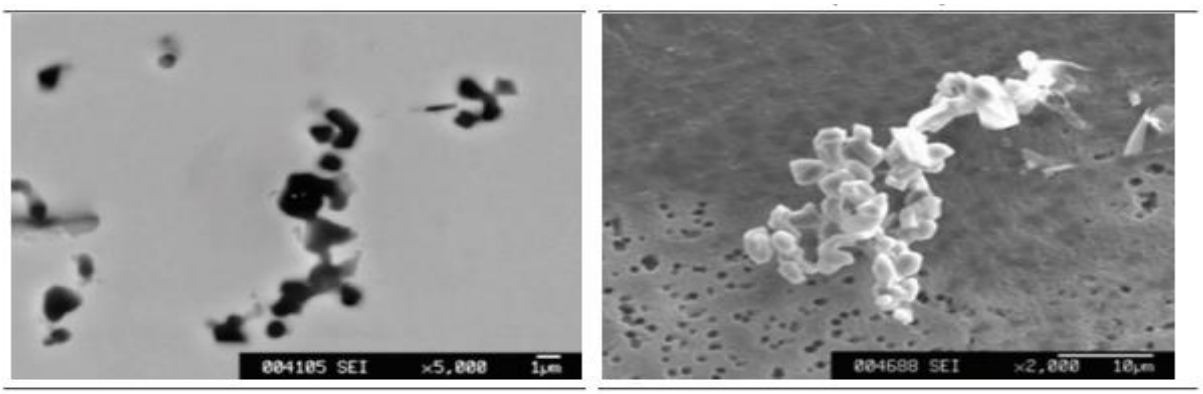


Figure 10: Both images shows a cluster of inclusions. Picture 1 shows a cross section of the metal sample, picture 2 shows the cluster after electrolytic extraction onto a film filter [31].

There is also a possibility that dendrites may appear on microscopic images. The shape of a dendrite in a 2D visualization is usually circular, which is a problem when classifying inclusions since the dendrites might be mistaken for inclusions.

3.4 Alumina inclusions and calcium treatment

Alumina inclusions tend to have a negative impact on the steels mechanical and processing properties. For example, the clustering of alumina tends to block the nozzle in a continuous casting process. The complex morphologies of the clusters also tend to increase the local stresses. To decrease the clustering calcium treatment is applied. The calcium transforms the alumina to calcium aluminate lowering the melting point of the inclusion to about 1350°C-1405°C, thus making the nozzle block problem disappear while casting since casting temperatures are usually higher. The calcium inclusions are usually around 5-10 [µm] in diameter. Equation 5 shows a typical formation of calcium aluminate from aluminium oxide and calcium [32] [33] [34].

$$[Ca] + 6\frac{1}{3}(Al_2O_3)_{inc} = (CaO.6Al_2O_3)_{Solid\ inc} + \frac{2}{3}[Al] \quad (5)$$

4 Method

This chapter will go through how to obtain the necessary data and the construction of a Convolutional Neural Network (CNN) to automate classification on two steel samples.

4.1 Data preparation

Two samples of steel were to be examined, the difference between them are the process parameters. Both samples have been exposed to calcium treatment. The difference between them is that one sample has been exposed longer to the treatment than the other. The longer treatment with calcium should increase the number and size of inclusions. The different samples were to be called 631 and 632. Due to corporate confidentiality, information about which steel grades and process parameters was not disclosed only the fact that calcium treatment had been performed.

To distinguish the two different steel samples based on inclusions, images had to be taken with high magnification at a high resolution. The method chosen to photograph the samples in this project is therefore with a Scanning Electron Microscope (SEM).

4.1.1 Sample preparation

The samples were moulded into samples made of PolyFast by Struers, a special conductive polymer made for SEM. The samples were then ground down on rotating disks with grinding papers consisting of different grit size in four steps, the grit size indicated how coarse a paper is the grit size is classified by ISO/FEPA grit designation. The first sand paper had a grit size of P240 meaning that the average particle diameter is about 58.5 [µm]. The secondary paper had P320 or a particle distance of 46.2 [µm]. The third paper had P600, 25.8 [µm] and the fourth and last paper had a fine grit size of 1200, 15.3 [µm].

After the grinding station the samples were cleaned, firstly with water and cotton cloth, and then with ethanol. After cleaning, the samples were dried using hot air.

Examination of the inclusions in the samples requires a polished surface. Polishing was performed after drying by adding diamond paste of 3 microns onto a rotating polishing disk. Lastly, the polished samples were washed with ethanol and directly after, dried with hot air. The samples are shown in figure 11.

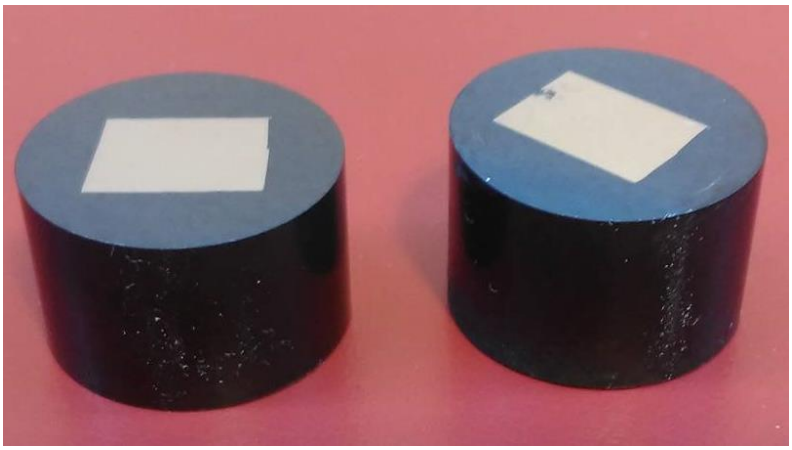


Figure 11: A picture showing the final versions of the two samples

4.1.2 SEM

To obtain pictures from the samples, the samples were placed inside the vacuum chamber of the SEM. The magnification was set to 250x, making inclusions visible at a size of approximately 8 µm, see figure 12. The type of electrons used for examination of inclusions was chosen to be Back-Scattered Electrons (BSE).

Figure 12: SEM picture describing the visibility of inclusions appearing as dark spots.

Pictures were taken manually, each picture was taken adjacent to the previous. To obtain a fine focus of inclusions in the two different samples, the light intensity parameter had to be changed, making the matrix of 631 slightly brighter than the matrix of 632. The contrast variance for the two samples is depicted in figure 13.

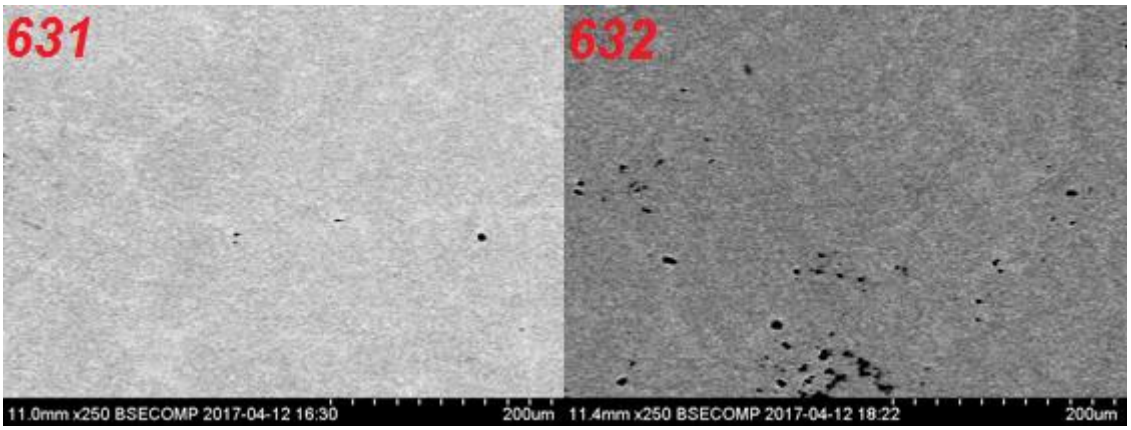


Figure 13: The contrast difference between sample 631 and 632.

The total pictures obtained from SEM was 417, 193 from sample 631 and 224 from sample 632, with a data size of 640x480 pixels and pixel size at 793.75. The acceleration voltage was set to 2000 [V], the magnification to 250x. The emission current was 66000 [nA] for sample 631, and 62000[nA] for sample 632. Working distance for sample 631 was 11.0 [mm] and 11.4 [mm] for sample 632.

4.1.3 Manual inspection

The obtained images for sample 631 and sample 632 was manually investigated comparing the amount of inclusions. The investigation was done to determine which of the samples that have undergone a longer calcium treatment.

4.1.4 Image processing

For the software to work as optimal as possible the dark underline in the images needed to be cut out. The reason for this operation is because dark and white areas create high-value gradients. As present time is shown and differs in each image, the underline may be hard for the software to handle. The underline of the images was removed, making the images size 640x448 pixels.

The images were resized to 448x448. The resizing was done because the network is easier to construct using square images. There is also a need to train the network on more than 417 images. Every image was cut into 16 images giving a total of 6672 images. The final size of the images were 112x112 pixels.

The images were then divided into three groups, one group for training consisting of 3300 (1650 of sample 631 and 1650 of sample 632) images, one group for validation consisting of 1650 (825 of sample 631 and 825 of sample 632) images and one group for testing with 1650 (357 of sample 631 and 1293 of sample 632) images. The training and validation images were set to the same amount for both samples to ensure a balanced training. The data was split into

multiples of 50 with 72 images discarded. It is a common approach used to ensure that the size of batches for training is the same [36].

A test with binarization of the pictures was made as well. The binarization converts the image to a black and white image. A threshold value of 50 on the grey scale was used, where 0 is black and 255 is white. The values over 50 convert to white while values under 50 are changed to black. Images with no inclusions transformed into only white images. The images that were completely white were removed since the purpose is to classify with regards to inclusions. The amount of images were set to the same number for both samples, to ensure a balanced training of the two samples. The number of training images was set to 644 for both sample 631 and 632, the number of validation pictures where for each sample set to 321 and the number of test images was set to 151 for each sample. The purpose with binarization was to evaluate how the substrates interact with the software. However, when binarizing vital information regarding the inclusions is lost such as colour contrast. Figure 14 shows a SEM image from sample 632 before and after binarization.

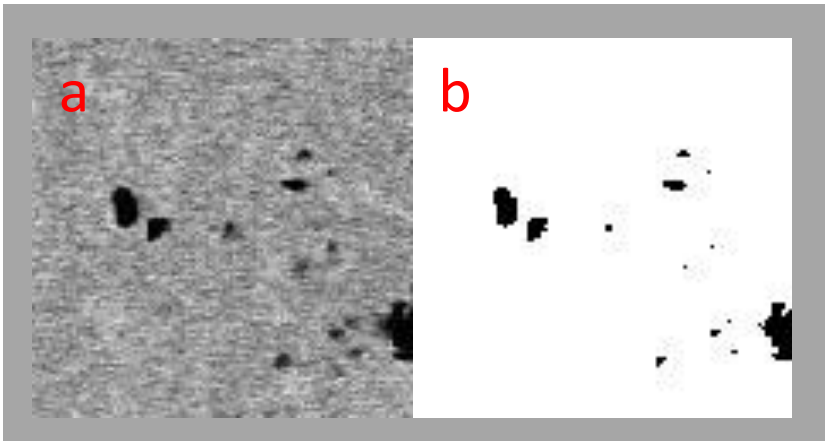


Figure 14: The difference in SEM image in sample 632 before binarization (a) and after binarization (b).

4.2 Creating the model

In this section, it will be described how the network was constructed and how the training was conducted.

4.2.1 Implementation

The software was built using the open source neural network application programming interface (API) provided by Keras Documentation [37], which is written in Python programming language. The Keras API enables users to build networks using different kinds of open source libraries for neural networks calculations such as Tensorflow [38] and Theano [39]. In this study, Tensorflow will be applied as the backend that deals with the mathematical aspect of the neural network.

The model was constructed as follows:

1. A convolutional 2-D layer containing 6 filters, with a kernel size of 7x7, striding with 1 and Rectified Linear Unit (ReLU) as the activation function is added. These settings decrease the input matrix of 112x112 down to a size of 106x106. The six filters produce six feature maps of size 106x106

2. The next layer added is a pooling layer that uses the max pooling as activation function, a filter of 2x2 decreases the size of the six feature maps down to 53x53.

3. Another convolution layer was added. This layer uses 16 filters, with a kernel size of 5x5, striding is again set to 1 and the activation function is again ReLU. Narrowing the matrix down to 49x49. In this layer 16 feature maps are obtained.

4. A second max pooling layer with a 2x2 filter reduces the size of the 16 feature maps to 24x24 is added.

5. A third convolution layer is then added with 16 filters and for this layer a 3x3 kernel. Decreasing the matrix dimensions down to 22x22. 16 feature maps are obtained.

6. A third and last max pooling layer is added, with a filter of again 2x2, reducing the 16 feature maps to a size of 11x11.

7. A flattening layer is added, which flattens the 16 filters of size 11x11 down to a one dimension vector consisting of 1936 neurons.

8. The last layers in the model are three Fully Connected (FC) layers. The first one consists of 60 neurons and the second consists of 30 neurons. Both use ReLU as an activation function. The last layer is a classification layer consisting of just two neurons, this layer is using softmax as the activation function.

The architecture of the network is shown in figure 15. The arrow connections indicate that both forward- and backpropagation is possible.

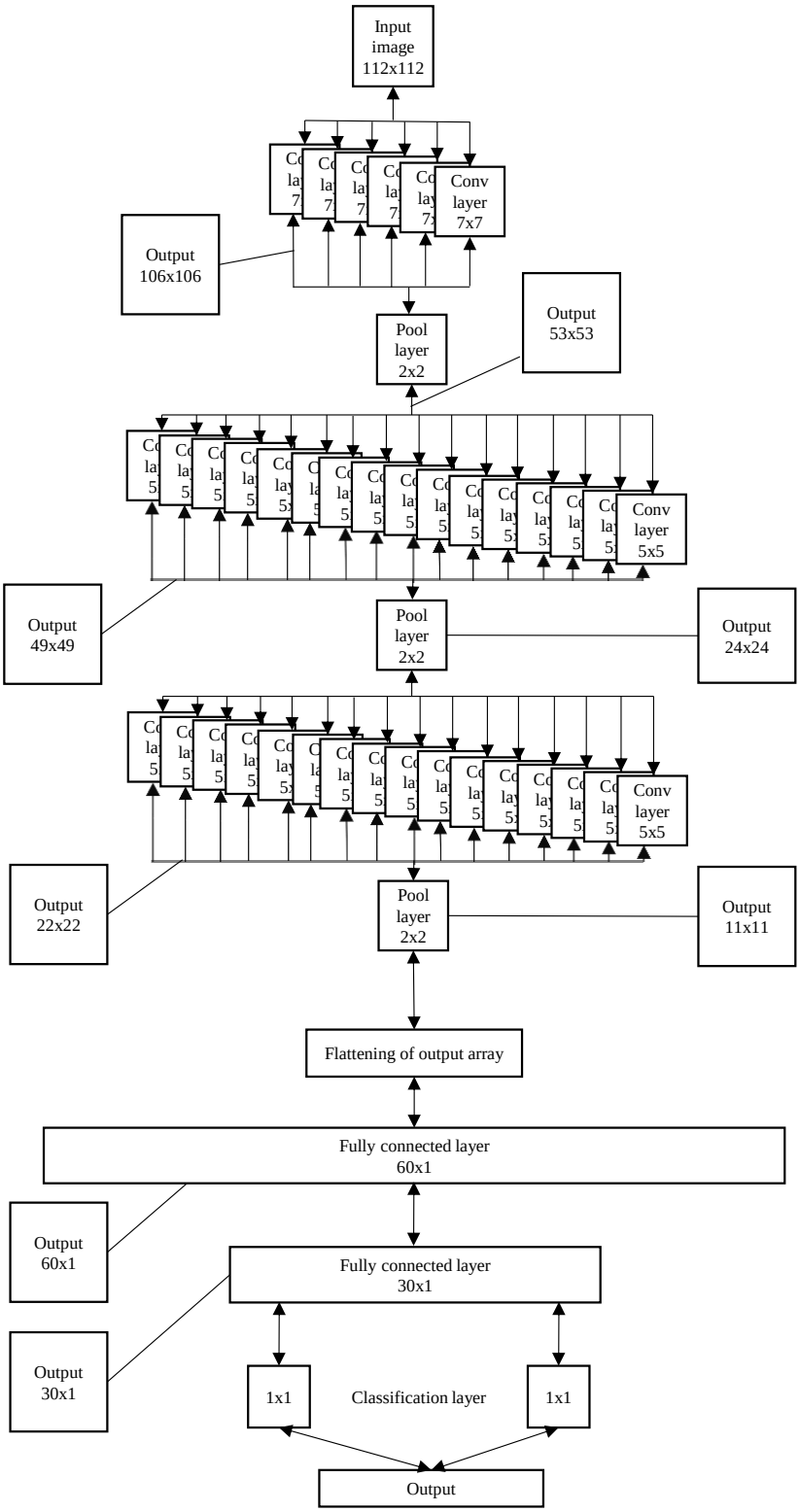


Figure 15: The constructed network, with connections and outputs from different layers. The arrows direction indicates the possibility of both forward propagation and backpropagation. Each square in the convolutional layer corresponds to a filter that creates a feature map.

4.2.2 Training, validation and testing

The networks initial weights were set by the Keras API to random low numbers, thus ensuring no initial saturation. The specific values and distribution of the random numbers were not known.

For the non-binary images, training consisted of 10 epochs with validation after each in order to supervise the improvement of the network. In every epoch, a batch size of 50 was set, meaning that the program takes 50 images before each backpropagation and weight update. When processing the binary images, three different amount of epochs was computed, 10, 200 and 1000 epochs. The learning rate for the network was predetermined by the Keras API.

During the training and validation, the accuracy and loss (error) were noted in order to evaluate the networks improvement over time. After the epochs of training and validation, the test data was introduced to the network. The results obtained were noted.

5 Results

5.1 Non-binary images

The improvement of the model and results obtained with non-binarized images will be presented in this section.

5.1.1 Manual inspection of the images

A manual analysis comparing the two steel samples, considering the inclusions, shows that steel sample 632 contained more inclusions than 631. The higher amount of inclusions indicates that sample 632 was the sample treated with calcium for a longer duration.

5.1.2 Training results

The results for the networks improvement during training and validation, with 3300 training samples and 1650 validation samples over 10 epochs is shown in figure 16.

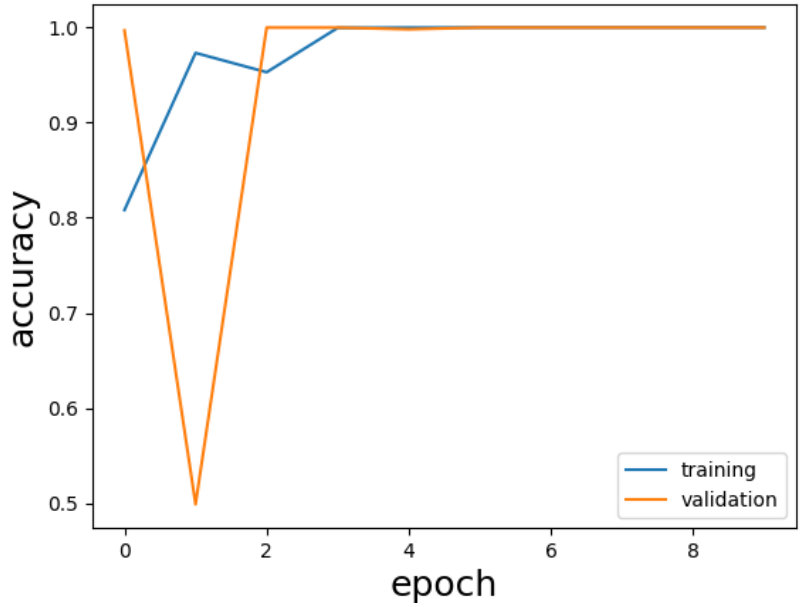


Figure 16: The networks accuracy improvement for the training set and validation set for each epoch.

The change in loss over the epochs for training and validation is shown in figure 17.

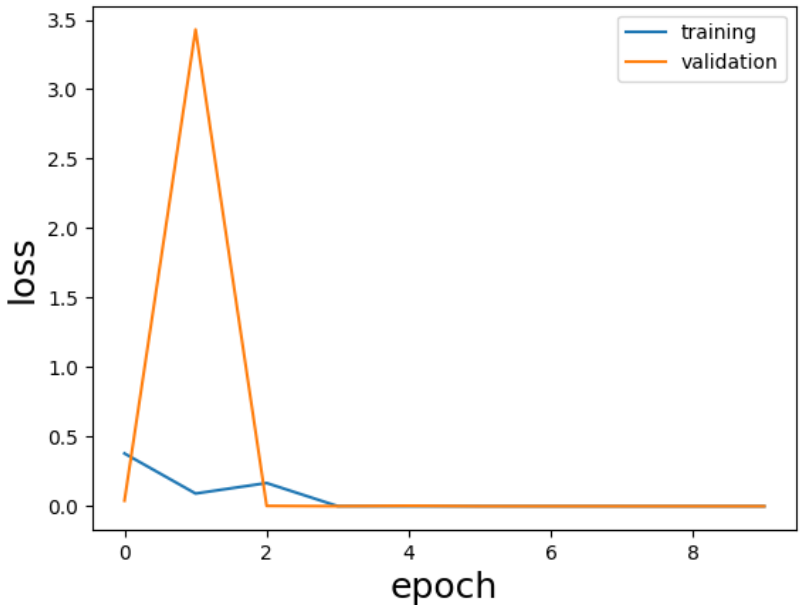


Figure 17: The networks decrease in loss for training set and validation set over 10 epochs

5.1.3 Test Results

The accuracy obtained from the test data, 1762 images, classifies the images correctly with 99.99 % probability.

5.2 Binary images

The improvement of the network and results obtained with binarized images will be presented in this section.

5.2.1 10 epochs

The networks accuracy improvement during training and validation, with 644 training images and 321 validation images is depicted in figure 18.

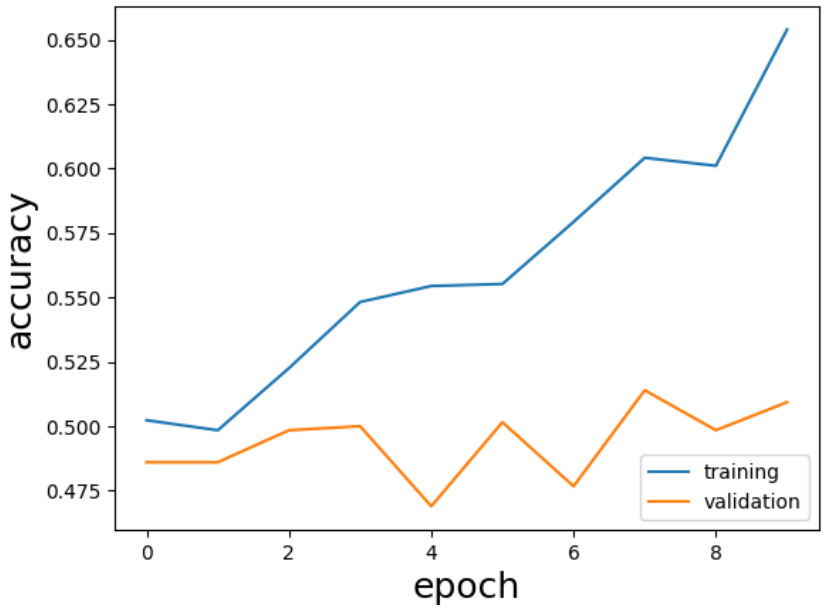


Figure 18: The improvement in accuracy for training and validation data for 10 epochs.

The change in loss over 10 epochs for training and validation is shown in figure 19.

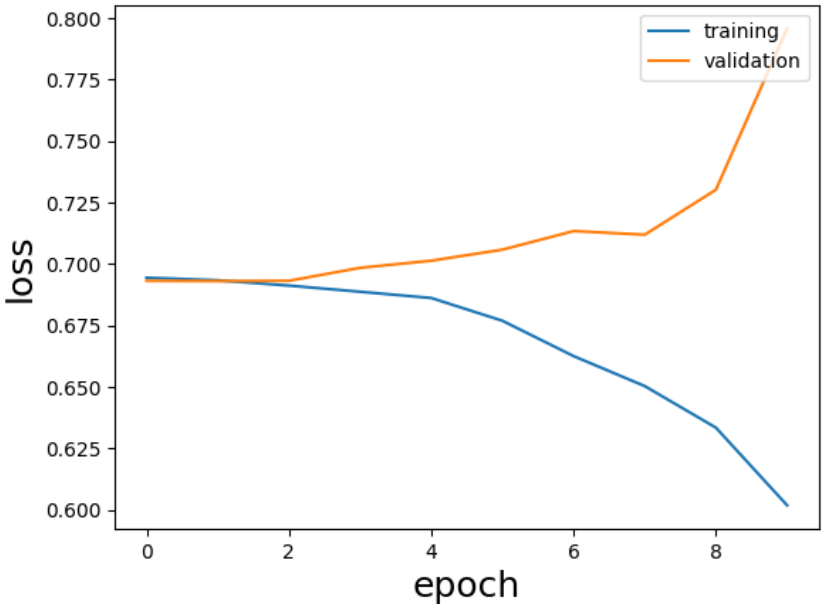


Figure 19: The decrease in loss over 10 epochs for training and validation data.

The accuracy obtained from the test data, 302 images, classifies the images correctly with 52.3 % probability.

5.2.2 200 epochs

The networks accuracy improvement during training and validation, with 644 training images and 321 validation images is depicted in figure 20.

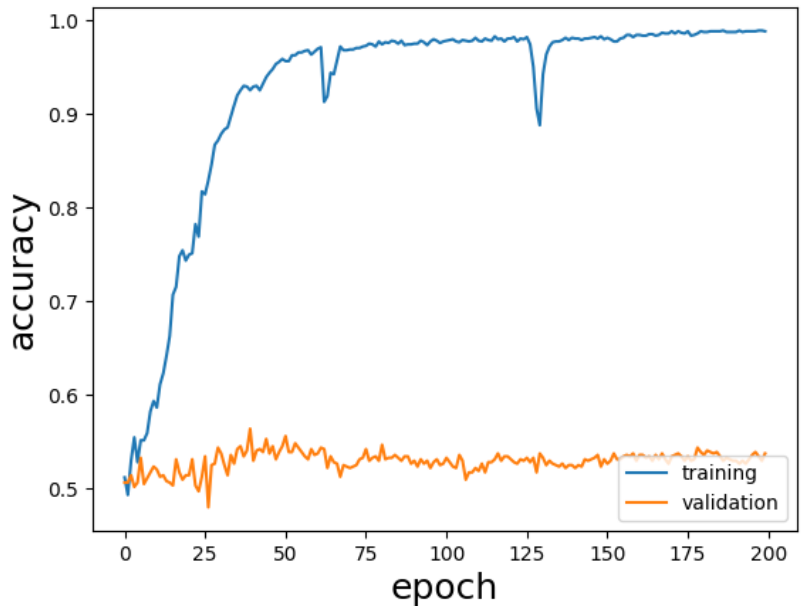


Figure 20: The improvement of accuracy over 200 epochs for training and validation set.

The change in loss over 200 epochs for training and validation is shown in figure 21.

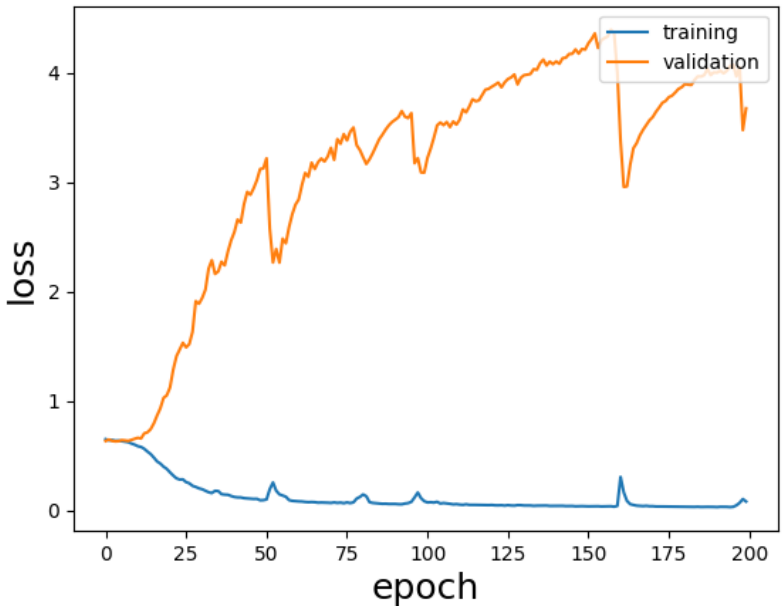


Figure 21: The change in loss over 200 epochs for training and validation.

The accuracy obtained from the test data, 302 images, classifies the images correctly with 67.9 % probability.

5.2.3 1000 epochs

The networks accuracy improvement during training and validation, with 644 training images and 321 validation images is depicted in figure 22.

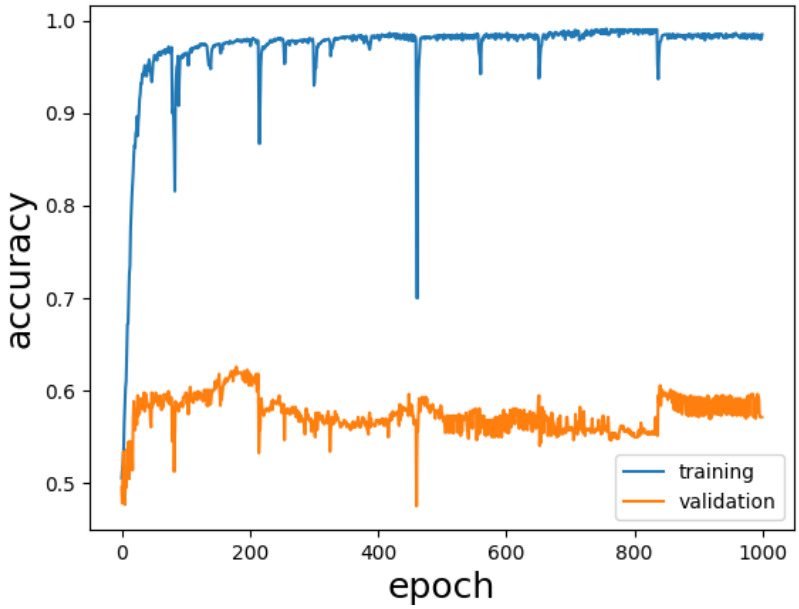


Figure 22: Accuracy change over 1000 epochs for training and validation data.

The change in loss over 1000 epochs for training and validation is shown in figure 23.

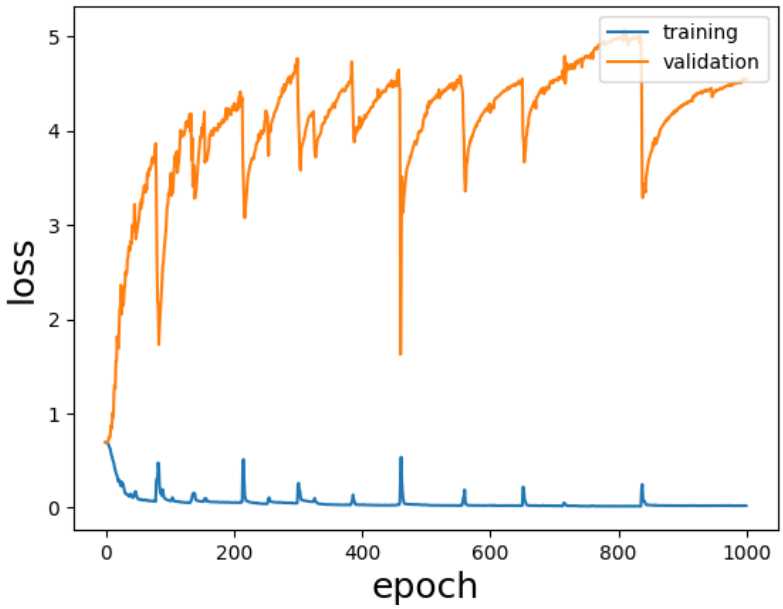


Figure 23: The change in loss over 1000 epochs for training and validation data.

The accuracy obtained from the test data, 302 images, classifies the images correctly with 61.2 % probability.

6 Discussion

The information about the process parameter for the steel samples was not known. The first thing to do was to examine the images taken in Scanning Electron Microscope (SEM) to compare the different samples and come to a conclusion on which sample was calcium treated for a longer time. The images show that for sample 632 the number of inclusion was greater than for sample 631, thus the conclusion was made that sample 632 was the one treated for a longer time. Since the inclusion appears to be spherical and the known fact that the samples have undergone calcium treatment, the assumption was made that all the inclusion were calcium based. The inclusion could be of a different composition. However, it is not relevant to this study.

6.1 Improvement of the network

The improvement of the network shown in figure 18 indicates that the model is learning. Both for the training set data and validation set data the network achieves high prediction accuracy. In both figure 16 and figure 17, the big variation in accuracy and loss for the first epochs can be explained by the randomly set weights for the first iteration during training.

The thesis main target was to classify two steel samples using machine learning techniques. The constructed program did achieve a classification on non-binary images with an accuracy of 99.99 % on the test data. The results show that machine learning models using Convolutional Neural Networks (CNNs) are more than capable of image recognition and classification of steel samples. The belief is that the high classification accuracy is due to the colour intensity of the substrates in the images for the two different samples. With light-grey substrate for sample 631 respectively dark-grey for sample 632. Therefore, trials with the images binarized were computed.

The binarization of the images forces the network to classify the steel samples only based on the inclusion size, form, and location. The trials with binary images started out at 10 epochs. The prediction accuracy obtained was only 52.3%. The low accuracy rate indicates that the network is not able to predict as it merely randomly distributes into the two categories. The theory was that either the number of epochs or learning rate was not enough to properly train the network. However, a too high learning rate can cause problems whilst converging. Therefore, further tests were done with 200 and 1000 epochs with the same learning rate to analyze if the prediction accuracy increases. Both trials with more epochs gave results on prediction accuracy over 60 %, with the highest prediction rate at 67.9 % for 200 epochs. This clearly shows that the number of epochs impacts the learning of the network. The difference between 200 and 1000 epochs shows that adding more epochs will provide better prediction up to a certain point. The validation accuracy improvement in figure 22 shows that maximum accuracy for validation data is reached at around 800 epochs.

Plausible explanations for the peak in accuracy at 800 epochs is that the amount of data restricts the training and that the network is quite shallow. Possible changes to the network for better predictions could include constructing a deeper network, i.e. more hidden layers, have a different topology, for example, two convolutional layers before the first pooling layer to extract more distinct features of the image data or acquire more data.

A thorough investigation of the improvement regarding the network, whilst processing binary images indicates that the network could have some problems with overfitting. The network gets high accuracy rates for training data, meaning that it generalizes the image data well. But for the validation and testing data, the network cannot achieve a generalization of the data enough for high prediction accuracy. This is also indicated by the loss over epoch graphs. Ideally, the loss should decrease over time, not increase. However, the pooling layers purpose is to reduce the overfitting problem. A possible explanation for the overfitting problem is that the pooling layers destructive impact on the data removes too much information from the validation set. The same reasoning ought to apply to the training set, but the training set contains more images (data) and therefore probably not affected on the same scale.

The binarization of the images may, as stated earlier, remove vital information regarding the inclusions. Therefore trials were the image substrate is normalized should be conducted. The normalization will provide a similar contrast of the substrates for the different samples rather than changing the whole image, thus preserving the vital inclusion information. Approaching the problem from this angle could give the opportunity to classify specific inclusions.

6.2 Economic, social and environmental aspects

If automated classifiers were integrated into the steel industry, it could reduce the amount of time spent on classification drastically. Time previously spent on classification can be used for other important assignments in the steelmaking industry. The investment in computer power and software necessary would probably be insignificant compared to the manpower cost over time. A fully functional CNN model would therefore be profitable for the companies.

The implementation will require that companies acquire the competence necessary for the process. Acquiring of the competence can be achieved either by supplementary training of pre-existing staff, employment of new personnel or on a consulting basis.

The automated classification would aid experts in the steelmaking industry when image analysis is necessary. It could provide information that can help the experts make better and faster decisions regarding material analysis. The decisions could lead to a more efficient production and lower material consumption. More efficient production will result in a lowering of environmental impact.

7 Conclusions and recommendations

The conclusion of this study is that Convolutional Neural Networks (CNNs) is a powerful tool for classification of steel samples by image recognition.

The classification performed by the network is very good for the non-binarized images. However, to implement this on other images taken with more similar substrates, the classification must be done regarding the inclusions. The binarized images give the possibility to disregard the substrate. However, further studies are needed before possible implementation in the steel industry.

To realize the implementation, work should be done on images where the substrate is normalized instead of binarized. To achieve implementation in the steel industry a substantial amount of data is required for the training. Also, investigations on different topology networks will determine which pattern of layers that can provide the best predictions.

8 Further studies

Interesting areas in which Convolutional Neural Networks (CNN) can be applied for further studies is presented below:

- Using images of different types of inclusions with regards to morphology and quantity. To facilitate the possibility to classify specific inclusions based on images instead of conventional chemical analysis.

- Investigate the performance of deeper networks and with different topologies with the goal to attain higher prediction rates.

- Normalizing existing images in an attempt to attain a similar contrast of the substrate. Train a CNN on that data to see the differences in accuracy compared to the models in this study.

9 Acknowledgements

We would like to express our heartfelt thanks to our supervisor, Leo Carlsson, Enheten Processer, KTH for the support and feedback throughout this project.

Our sincere thanks to associate Prof. Peter Hedström, Enheten Strukturer, KTH and Wenli Long, Avancerade Instrument Materialvetenskap, KTH for the invaluable help in obtaining images via SEM and sample preparation.

Our grateful thanks to Andrey Karasev, Enheten Processer, KTH for providing the steel samples that the thesis was based on.

Finally, our gratitude to Prof. Pär Jönsson, Enheten Processer, KTH for the permission to use his photograph in figure 10.

10 References

[1] Y. Bi, “Three Dimensional Determinations of Inclusions in Ferroalloys and Steel Samples,” Royal Institute of Technology, Stockholm, 2014.

[2] L. Carlsson, “Using Multilayer Perceptrons as means to predict the end-point temperature in an Electric Arc Furnace,” KTH Royal Institute of Technology, Stockholm, 2015.

[3] J. Mesa, C. Menendez, F. Ortega and P. Garcia, “A smart modelling for the casting temperature prediction in an electric arc furnace,” *International Journal of Computer Mechanics,* vol. 86, no. 7, pp. 1182-1193, 2009.

[4] H. Ma, W. You and T. Chen, “Prediction Model of Molten Iron Endpoint Temperature in AOD Furnace based on RBF Neural Network,” in *International Conference on Logistics Systems and Intelligent Management*, 2010.

[5] K. Liao and Y. Lee, “Detection of rust defects on steel bridge coatings via digital image recognition,” *Automation in Construction,* vol. 71, pp. 294-306, 2016.

[6] Y. Tang, X. Zhang, X. Li and X. Guan, “Application of a new image segmentation method to detection of defects is castings,” *The International Journal of Advanced Manufacturing Technology* , vol. 43, no. 5, pp. 431-439, 2009.

[7] R. Pidaparti, B. Aghazadeh, A. Whitfield, A. Rao and G. Mercier, “Classification of corrosion defects in NiAl bronze through image analysis,” *Corrosion Science,* vol. 52, no. 11, pp. 3661-3666, 2010.

[8] S. Chen, B. Lin, X. Han and X. Liang, “Automated inspection of engineering ceramic grinding surface damage based on image recognition,” *International Journal of Advanced Manufacturing Technology,* vol. 66, no. 1, pp. 431-443, 2013.

[9] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, 2012.

[10] Y. Adachi, M. Taguchi and S. Hirokawa, “Microstructure Recognition by Deep Learning,” *Tetsu-to-Hagane,* vol. 102, no. 12, pp. 722-729, 2016.

[11] L. Yi, G. Li and M. Jian, “An End-toEnd Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks,” *Steel Research International,* vol. 88, no. 2, 2017.

[12] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber and G. Fricout, “Steel Defects Classification with Max-Pooling Convolutional Neural Networks,” in *WCCI 2012 IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 2012.

[13] S. Zhou, Y. Chen, D. Zhang, J. Xie and Y. Zhou, “Classification of surface defects on steel sheet using convolutional neural networks,” *Materiali in Tehnologije,* vol. 51, no. 1, pp. 123-131, 2017.

[14] S. Lee and S. Kim, “Localization of the slab information in factory scenes using deep convolutional networks,” *Expert Systems with Applications,* vol. 77, pp. 34-43, 2017.

[15] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[16] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning (2nd edition), Springer-Verlag, 2009.

[17] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning, with Applications in R, New York: Springer, 2015.

[18] C. Bishop, Pattern Recognition and Machine Learning, New York: Springer, 2006.

[19] K. Priddy and P. Keller, Artificial Neural Networks: An Introduction, Bellingham, Washington USA: SPIE PRESS, 2005.

[20] K. O`Shea and R. Nash, "An Introduction to Convolutional Neural Networks," *Computing Research Repository,* vol. arXiv: 1511.08458, no. 2, 2015.

[21] M. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *Computing Research Repository,* vol. arXiv: 1311.2901, no. 3, 2013.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *Computing Research Repository,* vol. arXiv:1512.03385, no. 1, 2015.

[23] A. Shepherd, Second-order methods for neural networks: fast and reliable training methods for multi-layer perceptrons, London: Springer-Verlag Limited, 1997.

[24] y. Tang, "Deep Learning using Linear Support Vector Machines," *Computing Repository Research,* vol. arXiv: 1306.0239, no. 4, 2015.

[25] R. Kiessling and N. Lange, Non-metallic inclusions in steel / P.3, London: Iron and steel institute, 1968.

[26] P. Beeley, Foundry Technology, Oxford: Butterworth-Heineman, 2001.

[27] A. Stiénon, A. Fazekas, J. Buffière, A. Vincent, P. Daguier and F. Merchi, "A new methodology based on X-ray micro-tomography to estimate stress concentrations around inclusions in high strength steels," *Materials Science and Engineering: A,* Vols. 513-514, no. 1, pp. 376-383, 2009.

[28] K. Yamamoto, H. Yamamura and Y. Suwa, "Behaviour of Non-metallic Inclusions in Steel during Hot Deformation and the Effects of Deformed Inclusions on Local Ductility," *ISIJ International,* vol. 51, no. 12, pp. 1987-1994, 2011.

[29] S. S. Chaeikar, "Examination of inclusion size distributions in duplex stainless steel using electrolytic extraction," Royal Institute of Technology, Stockholm, 2013.

[30] M. Safa, "3D study of non-metallic inclusions by EE method and use of statistics for the estimation of largest size inclusions in tool steel," The Royal Institute of Technology, Stockholm, Sweden, 210.

[31] D. Janis, R. Inoue, A. Karasev and P. Jönsson, "Application of Different Extraction Methods for Investigation of Nonmetallic Inclusions and Clusters in Steels and Alloys," *Advances in Materials Science and Engineering,* vol. 2014, p. 7, 2014.

[32] M. H. M. L. &. M. L. L. Holappa, "Thermodynamic examination of inclusion modification and precipitation from calcium treatment to solidified steel," *Ironmaking & Steelmaking,* vol. 30, no. 2, pp. 111-115, 2013.

[33] M. &. H. L. Lind, "Transformation of Alumina Inclusions by Calcium Treatment," *Metallurgical and Materials Transactions B,* vol. 41, no. 2, pp. 359-366, 2010.

[34] Z. L. W. X. R. Y. L. X. S. Q. Yang W., "Characteristics of Inclusions in Low Carbon Al-Killed Steel during Ladle Furnace Refining and Calcium Treatment," *SIJ International,* vol. 53, no. 8, pp. 1401-1410, 2013.

[35] L. Carlsson, Interviewee, *PHD Student KTH.* [Interview]. 20 April 2017.

[36] "Keras Documentation," [Online]. Available: keras.io. [Accessed 2017].

[37] "Tensorflow," [Online]. Available: www.tensorflow.org. [Accessed 2017].

[38] "Theano," 21 April 2017. [Online]. Available: http://deeplearning.net/software/theano/. [Accessed 22 May 2017].