



Chair of Ferrous Metallurgy

Master's Thesis

Influence of Different Hyperparameter
Settings and Data Preprocessing Methods
on the Classification of Nonmetallic
Inclusions with Machine Learning
Algorithms

Robert Musi, BSc

April 2023



Herrn **Robert MUSI** wird vom Lehrstuhl für Eisen- und Stahlmetallurgie folgendes Masterarbeitsthema S748 gestellt:

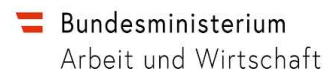
Influence of Different Hyperparameter Settings and Data Preprocessing Methods on the Classification of Nonmetallic Inclusions with Machine Learning Algorithms

Im Rahmen der automatisierten REM/EDX Analyse werden sowohl morphologische Daten (ECD, mittlerer Grauwert, Position, Form) als auch Rückstreuungsbilder von nichtmetallischen Einschlüssen erzeugt. Der Grauwert in diesen Bildern weist dabei einen Zusammenhang mit der chemischen Zusammensetzung des Einschlusses auf und gemeinsam mit der Grauwertverteilung können Rückschlüsse auf die Einschlussklasse gezogen werden. Ziel der Masterarbeit ist es, unterschiedliche Ansätze des maschinellen Lernens für die Klassifikation von nichtmetallischen Einschlüssen anhand der morphologischen Daten und der Grauwertbilder zu untersuchen und zu optimieren.

Inhalte der Arbeit:

- Literaturrecherche
- Evaluierung unterschiedlicher Algorithmen des maschinellen Lernens und Vergleich der Ergebnisse anhand diverser Kennzahlen und Grafiken
- Detaillierte Auswertung und Aufbereitung des vorhandenen Datensatzes
- Einfluss unterschiedlicher Hyperparameter (z.B. Tiefe des Netzwerks, Anzahl an Neuronen in den einzelnen Schichten, Aktivierungsfunktionen) und des Datensatzes auf die Ergebnisgenauigkeit
- Diskussion, Vergleich und Zusammenfassung der Ergebnisse

Industriepartner:



Wir bedanken uns für die finanzielle Unterstützung durch das Bundesministerium für Arbeit und Wirtschaft und die Nationalstiftung für Forschung, Technologie und Entwicklung sowie die Christian Doppler Forschungsgesellschaft.

Leoben, April 2023

Assoz. Prof. Dr. Susanne Michelic



EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 20.04.2023

Unterschrift Verfasser/in
Robert Musi

Acknowledgement

First and foremost, I would like to thank my thesis supervisor, Susanne Michelic, for her guidance, contribution, and invaluable feedback through the entire time of this thesis.

I am also grateful to the members of the Christian Doppler Laboratory at the Chair of Ferrous Metallurgy. At this point, I would like to mention Kathrin Thiele for her constructive advice and guidance through various SEM measurements, and Shashank Ramesh Babu for interesting discussions about machine learning and data-related topics.

In the last words of the acknowledgement, I would like to express my appreciation to all my friends, and most importantly, my family, for their financial support and motivation throughout my entire time at the university.

Kurzfassung

Die automatisierte Rasterelektronenmikroskopie mit energiedispersiver Röntgenspektroskopie (REM/EDX) ist eine etablierte Methode zur Analyse von nichtmetallischen Einschlüssen (NME), die sowohl in der Forschung als auch in der Industrie Einsatz findet. Obwohl diese Methode eine detaillierte Auswertung von NME ermöglicht, ist der entscheidende Nachteil der Zeit- und Arbeitsaufwand, der für eine Probe bis zu mehreren Stunden in Anspruch nehmen kann. Methoden des maschinellen Lernens, die Daten sehr schnell verarbeiten, bieten eine interessante Alternative für die Charakterisierung von Einschlüssen. Die vorliegende Arbeit befasst sich mit der zeit- und energieeffizienten Klassifizierung von nichtmetallischen Einschlüssen, die durch das Training von Algorithmen des maschinellen Lernens mit Bilddaten von der automatisierten REM/EDX Analyse durchgeführt wurde. Für das Training und für die Evaluierung der Algorithmen kamen verschiedene Features, extrahiert von Rückstreuelektronenbildern ausgehend von sieben unterschiedlichen Stählen, zur Anwendung. Die Methoden des maschinellen Lernens aus den Python Bibliotheken scikit-learn und PyTorch wurden verwendet. Die höchste Trefferquote von 73,1 % erreichte das Random Forest Klassifikationsverfahren durch Training mit den Grauwerten der Bildpixel. Neuronale Netze konnten nicht auf diesem Niveau klassifizieren. Die verwendete Featureart, der Algorithmus und die geometrischen Abmessungen von NME hatten einen signifikanten Einfluss auf die Performance bei der Klassifizierung. Für weiterführende Studien müssen Parameter wie die Einteilungskriterien der NME bei der Datenvorverarbeitung, als auch Helligkeits- und Kontrasteinstellungen bei der automatisierten REM/EDX Analyse miteinbezogen werden, um die Trefferquote weiter zu verbessern.

Abstract

The automated scanning electron microscopy with energy dispersive X-ray spectroscopy (SEM/EDS) is a state-of-the-art method for the analysis of non-metallic inclusions (NMI) and well established for research and industry applications. Even though NMIs can be evaluated thoroughly with this method, the crucial disadvantage is its time effort, taking up to several hours per sample. Machine learning algorithms offer an interesting alternative for inclusion characterization, as data can be processed very fast. The present work deals with the time and energy efficient classification of non-metallic inclusions, carried out through the training of different machine learning algorithms on automated SEM/EDS generated image data. Features extracted from backscattered electron images, which were generated through the automated SEM/EDS analysis of seven different steels, served the purpose of training and evaluating the machine learning models. Various algorithms from the python libraries scikit-learn and PyTorch were used. The highest accuracy score of 73,1 % could be achieved with the Random Forest classifier trained on gray values of image pixels. Neural networks were not as suited for inclusion characterization. The used features, type of algorithm, and the NMI dimensions significantly influenced the classification performance. For further studies, parameters such as labeling criteria for NMI within the data preprocessing as well as contrast and brightness settings during automated SEM/EDS measurement need to be adapted in order to enhance accuracy scores.

Content

Acknowledgement	I
Kurzfassung	II
Abstract	III
Content	IV
1 Introduction	1
2 Non-metallic Inclusions in Steel	2
2.1 Origin and Control of Inclusions in Steelmaking	3
2.2 Inclusion Characterization Methods	5
2.3 SEM/EDS Analysis	6
2.3.1 Automated SEM/EDS Measurement with Oxford Instruments' Aztec Software	7
2.3.2 Feature Evaluation Tool	8
2.3.3 Backscattered Electron Images	10
3 Data Science Methods for Inclusion Characterization	11
3.1 Principal Component Analysis on Inclusion Datasets.....	11
3.2 Machine Learning for Determining the Abundance of Inclusions.....	12
3.3 Classification of Non-metallic Inclusions	13
3.3.1 Geometric Data Based Classification	14
3.3.2 Image Data Based Classification.....	15
4 Data Preprocessing	17
4.1 Data Pipeline for Exporting SEM/EDS Data	17
4.2 Description and Statistical Overview of the Dataset	20
4.2.1 Class Distribution.....	21

4.2.2	Type Distribution.....	24
4.2.3	BSE Images.....	26
4.3	Training Data.....	29
4.3.1	Feature Definition.....	29
4.3.1.1	Vertical and Horizontal Gray Value Gradients.....	31
4.3.1.2	Statistical Parameter from Gray Value Histograms.....	33
4.3.2	Feature Selection of Geometric Parameters.....	36
4.3.3	Comparing Different Steels.....	37
5	Training and Evaluation of Machine Learning Models.....	40
5.1	Definition of the Training Sets.....	41
5.2	Definition of the Machine Learning Models.....	42
5.3	Comparison between Under- and Oversampling.....	43
5.3.1	Sampling Strategy 'not majority' and 'not minority'.....	44
5.3.2	Alternative Sampling Strategy.....	47
5.4	Comparison between Class- and Type Labels.....	48
5.5	Bagging Classifier and Random Forest Classifier.....	49
5.5.1	Comparison of the Performance between different NMI-Features.....	51
5.5.2	Fine-Tuning.....	51
5.5.2.1	Fine-Tuning of the Bagging classifier.....	52
5.5.2.2	Fine-Tuning of the Random Forest classifier.....	54
5.5.3	Feature Importance.....	56
5.5.4	ROC Curves.....	57
5.6	Influence of Inclusion Dimensions.....	59
5.7	PyTorch: Deep Learning in Python.....	61
5.7.1	Multilayer Perceptron.....	63
5.7.2	Convolutional Neural Network.....	65
6	Conclusion and Outlook.....	68
	Bibliography.....	71
	Acronyms.....	76
	List of Tables.....	77
	List of Figures.....	78
	Appendix.....	81
A.1	Chemical Composition of the Steels.....	81

1 Introduction

Data science plays a crucial role in today's industry. With the ongoing digitalization and exponential growth of generated data in recent years, the ability to collect, process, and analyze this data has become a key differentiator for companies in almost every industry. In manufacturing, the access to quality-related data enables long-lasting enhancement of process and product quality [1]. Data science techniques, such as machine learning and artificial intelligence, are used to automate repetitive tasks, increase efficiency, and uncover hidden patterns and relationships from given information. In the steel industry, machine learning receives an increasing attention for handling high-dimensional datasets with reasonable effort. Researchers have been focusing on using data-driven models to improve product quality [2–5], detect problems early in the process [2,6], and gain process knowledge to improve process stability [7,8]. Due to the increasing demands on steel performance for manufacturers, introducing machine learning techniques for steel cleanliness evaluations could be a helpful tool in fulfilling advanced requirements. The origin, behavior in steelmaking, and influence on mechanical properties of non-metallic inclusions (NMIs) is an important topic on which researchers and industry have dealt with for almost 100 years [9]. In 1918, A. McCance [10] presented a study demonstrating that non-metallic inclusions in steel have an important effect on its properties, particularly in producing defects and causing failures. Since then, different methods for inclusion characterization have been developed and established in the industry and scientific environment. As stated by Zhang et al. [11], accurate methods for evaluating steel cleanliness are crucial in order to analyze and regulate it effectively.

The automated scanning electron microscopy with energy dispersive X-ray spectroscopy (SEM/EDS) is a direct method for particle analysis on cross-section samples. It enables the measurement of morphology and chemical composition of non-metallic inclusions at

sub-micrometer to millimeter dimensions. Even though this method can be used for a wide range of research and industry applications with certain benefits, the crucial disadvantage is its time effort [9]. Due to the fast-processing speed of machine learning algorithms, this drawback can be nullified when they are trained on automated SEM/EDS measurement data and alternatively used for particle analysis. The time saving aspect could become valuable for enabling an on-line steel cleanliness control tool at different stages in steelmaking.

The present work deals with the time and energy efficient classification of non-metallic inclusions, carried out through the training of different machine learning algorithms on BSE image data. Chapter 2 gives an overview about inclusion characterization methods, highlighting the automated SEM/EDS measurement. A short introduction into how machine learning can be applied in the field of inclusion characterization is presented in chapter 3. Chapter 4 starts with a description and a detailed study on the non-metallic inclusion datasets, which were used to train the machine learning algorithms. Various morphology characteristics and the representation of NMIs in BSE images of different inclusion classes are being discussed thoroughly. Furthermore, this chapter deals with possible feature extraction methods to simplify the classification task. Chapter 5 contains the evaluation of machine learning models using different performance metrics. The Bagging classifier and Random Forest classifier are described in detail. Additionally, this chapter presents methods for enhancing the algorithm's performance, including fine-tuning of hyperparameters, and showcases the application of neural networks.

2 Non-metallic Inclusions in Steel

For over a century, researchers have been dealing with the origin, control, and characterization of non-metallic inclusions (NMI). The ongoing rise in demand for improved steel performance

has led to the development of increasingly complex steel production routes, resulting in the involvement of more research fields focusing on steel cleanliness. [9]

2.1 Origin and Control of Inclusions in Steelmaking

To prevent the formation of low melting compounds, such as FeO, FeS and their eutectics, during solidification, deoxidation and desulphurization are used in secondary metallurgy to decrease the oxygen and sulfur levels in steel. High oxygen affinity elements like Mn, Si, and Al act as deoxidizing agent and form non-metallic deoxidation products in the liquid melt. Desulphurization is achieved by slag refining, where most of the sulfur must be removed, and by forming of precipitates during solidification. Only elements such as Ca, Mg, or rear-earths with a low iron solubility have a sufficient high sulfur affinity to form non-metallic sulfides at liquid metal temperatures. Formation products of deoxidation and desulphurization lead to endogenous non-metallic inclusions. Contrary to oxidic and sulfidic NMIs, nitrides do not form in the liquid phase but precipitate during cooling of the solidified steel along the austenite grain boundaries. Al, Nb, Ti, V, Zr and B act as nitride forming elements. NMIs can also originate from external sources such as fragments of refractories or entrapped slag, which are referred to as exogenous inclusions. [12,13]

The following classification categories of NMIs are commonly used [12]:

- Chemical composition: sulfides, oxides, nitrides
- Origin: endogenous, exogenous
- Forming stage: primary (formed under isothermal conditions in liquid steel), secondary (formed during cooling to liquidus temperature), tertiary (formed between solidus and liquidus temperature), quaternary (formed in solid steel)
- Size: macro, micro (An inclusion size of 20 μm is often used as threshold. Da Costa e Silva [12] defines an inclusion as macro, if it is large enough to cause immediate failure of the product.)

For inclusion removal by slag, which is often carried out in the ladle furnace, tundish, or mold, three steps must be considered: flotation in the bath, separation at the steel/slag interface and dissolution in the slag. The flotation by bath stirring and rising bubbles ensures the transport of NMIs to the steel/slag interface. Near the steel/slag interface, non-metallic inclusions must bind securely to the slag on reaching the interface. Based on thermodynamics, inclusions release energy during breakthrough of the steel/slag interface. For this to happen, a hole between the NMI and the steel/slag interface is formed by draining out steel. This hole uses

interfacial energy to grow spontaneously, and non-metallic inclusions get absorbed by the steel/slag interface. A schematic representation of this process is shown in **Figure 2-1**. A key factor for the absorption is that the inclusion energy exceeds the interfacial energy, which separates the two liquids. When the energy difference is insufficient, re-entering of the NMI in steel can happen and a poor removal efficiency is the consequence. The dissolution of particles (solid inclusions) is far more difficult than of droplets (liquid inclusions) due to the limited solubility in slags. Influencing parameters are the physical and chemical characteristics of the system, the temperature gradient and the volume of the slag. [12,14]

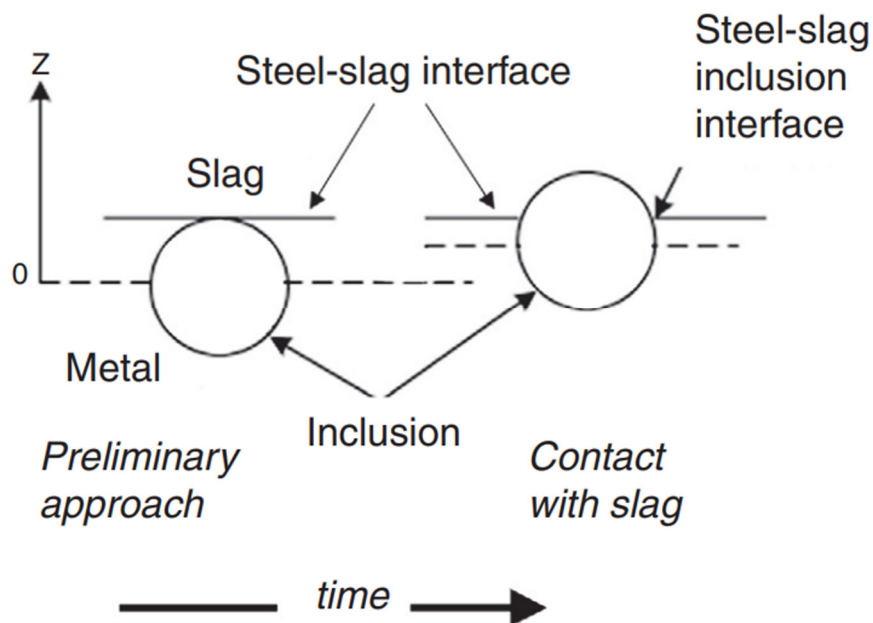


Figure 2-1: Schematic representation of an inclusion approaching and breaking through the steel/slag interface [12]

Controlling the amount, size, shape, morphology, and chemical composition of non-metallic inclusions in steel is essential for ensuring high cleanliness levels in the final product. The measured steel cleanliness can be correlated with the steel performance during production and application [9,11]. According to literature, steel properties can be positively and/or negatively affected by NMIs. The negative effects are mainly caused due to large or clustered particles [9] and can be summarized as follow:

- Mechanical anisotropy [15]
- Nozzle clogging during continuous casting [16,17]
- Influence on corrosion resistance (initiating pitting corrosion) [18]
- Fatigue behavior (acting as internal crack initiator) [19]
- Hot ductility (void formation) [20]

2.2 Inclusion Characterization Methods

Inclusion characterization is an important tool for studying the behavior, influence, and origin of NMIs during steelmaking and for measuring steel cleanliness at various process stages. The available methods can be divided into direct and indirect measurement techniques. Compared to direct methods, which are known for their accuracy but come at a higher cost, indirect methods offer a faster and more affordable solution, yet their reliability is limited to serving as relative indicators. [11]

The most established direct methods with their characteristic measurement properties are shown in **Figure 2-2**. In this work, the application of machine learning for inclusion characterization was enabled by using automated scanning electron microscopy with energy dispersive spectroscopy detectors (SEM/EDS) data. Therefore, the discussion of other methods is limited to a cursory overview:

- The optical emission spectrometry with pulse discrimination analysis (OES-PDA) uses a spark between an electrode and a metal surface, which is generated by electric energy. The vaporized atoms enter a high energy state within the discharge plasma. The PDA distinguishes and analyzes each single discharge. This method is faster than then the automated SEM analysis but does not provide any geometric information of the inclusions. [21,22]
- The Mannesmann Inclusion Detection by Analyzing Surfboards (MIDAS) is the most used ultrasonic testing (US) method for cast products. Steel samples are rolled to create surfboard samples and then scanned by ultrasonic to detect solid inclusions larger than 20 μm . [11]
- Silenos, abbreviation for steel inclusion level evaluation by numerical optical systems, is a method for steel cleanliness evaluation where layers with a thickness of 10 to 20 μm are removed from a disk-shaped sample by a CNC mill. A integrated high-resolution image scanner detects faulty areas on the surface, which are then evaluated by a laser spectrometer. [9,23]
- Inclusion characterization by optical microscope (OM) evaluation is the most common used method of the last decades. Since there is no direct measurement of the chemical composition, the inclusion type needs to be assumed by the morphology and grayscale in the images. International standards have been developed to determine the content of non-metallic inclusions. [9]
- Computed tomography (CT) techniques generate three-dimensional morphologies of inclusions through penetrating a steel sample with x-rays. NMI attenuate x-rays with a

different degree than the iron matrix. After ray detection and conversion to an electronic signal, a computer generates an image of the observed area. The morphology of complex clusters can be studied and evaluated, but no information regarding the chemical composition is obtainable. [24]

- Electrolytic or chemical extraction techniques includes dissolving the iron matrix and collecting noble, non-dissolving NMIs on a filter. After a sputtering process, the collected non-metallic inclusions are analyzed by SEM/EDS. This enables a characterization based on a three-dimensional morphology of the NMIs. [9]

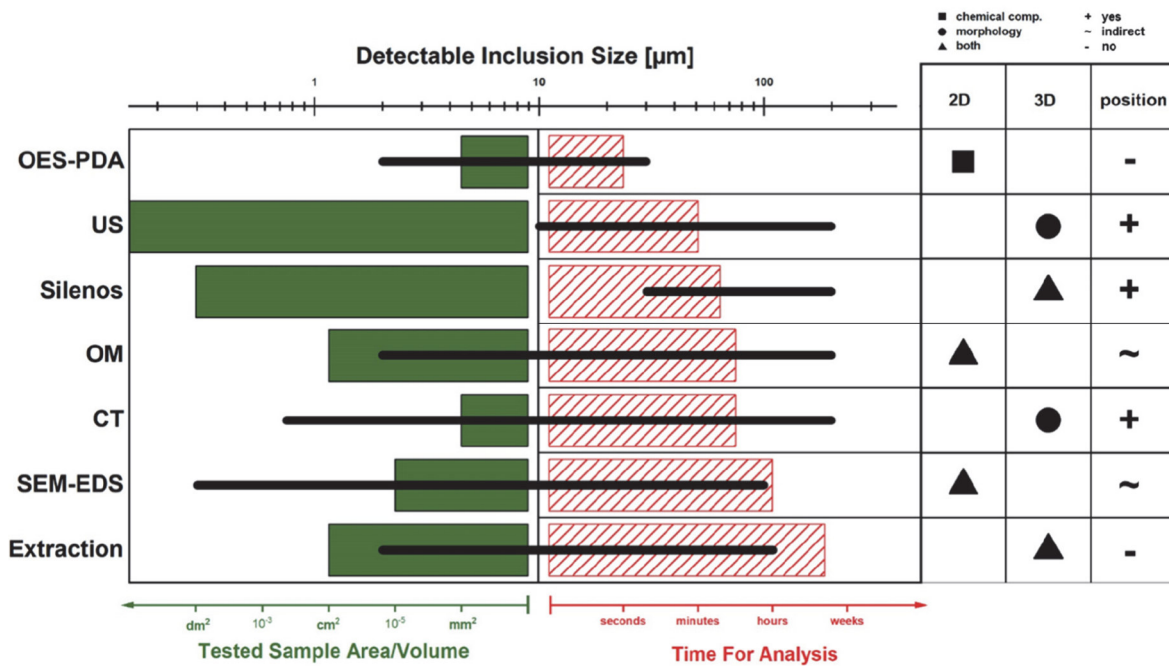


Figure 2-2: Comparison of the most common direct methods for steel cleanliness evaluation with measuring limits and characteristics [9]

2.3 SEM/EDS Analysis

A state-of-the-art method for inclusion characterization is scanning electron microscopy with energy dispersive spectroscopy detectors (SEM/EDS). A Schottky or field-emission cathode accelerates electrons through a voltage difference between a cathode and an anode. The voltage may be in the range of 0.1 keV to 50 keV. A two- or three-stage electron lens system demagnifies the electron beam so that an electron probe with a diameter of 1 to 10 nm is formed at the specimen’s surface. The final signal used for image formation is a result of a

combination of different atomic interaction processes. Secondary electrons, backscattered electrons (BSE), and Auger electrons form the emitted energy spectrum. For elemental analysis, either a wavelength- (WDS) or energy-dispersive spectrometer (EDS) is used. The WDS measures the wavelength of characteristic x-ray lines, while the EDS measures their quantum energy. For the analysis of non-metallic inclusions by EDS, a standard-based and a non-standard-based approach can be applied. The standard-based approach delivers the best analytical results because the magnitude of the characteristic signal level relative to the continuum defines the limits of quantification. A significant time and knowledge effort for composition analysis and the availability of standards for all elements measured in the sample material are the drawbacks of this method. Non-standard-based analysis are generally used for particle analysis due to the complicated quantification of complex multi-phase non-metallic inclusions. [9,25]

SEM/EDS analysis offers a wide range of application in research and industry. With magnifications of 25x to 150.000x, macroscopic ($> 100 \mu\text{m}$) and sub-microscopic ($< 1 \mu\text{m}$) NMIs can be evaluated. The automation of SEM/EDS is designed for rapidly analyzing steel samples containing information of every measured particle and generating statistically comprehensible data. Sample preparation contains generally embedding the steel in a conductive material (Cu) followed by grinding with SiC abrasive paper at various roughness levels (320-600-800-1200) and polishing at 9, 3 and 1 μm with a diamond suspension. Depending on the chemical composition of the sample, preparation steps need to be adjusted accordingly. After a cleaning and drying process, the sample can be analyzed with SEM/EDS. [9]

2.3.1 Automated SEM/EDS Measurement with Oxford Instruments' Aztec Software

Within this work, steel samples were analyzed by automated SEM/EDS measurement with a field-emitter-based JEOL 7200F and the operating software Oxford Instruments' Aztec at the Chair of Ferrous Metallurgy. For the automated process, microscope and measurement parameters are defined once at the beginning. The particle detection and identification are then controlled by the software without manual interference. To distinguish particles from the surrounding matrix, a grayscale threshold is defined by the user. As stated by Mayerhofer [9], microscope settings such as contrast and brightness should be adjusted until a mean matrix value of >24000 and a NMI value of <2000 is measured during image calibration. The mentioned values refer to a 16-bit grayscale image. **Figure 2-3** shows the calibration of an automated SEM/EDS measurement with a multi-phase NMI. Inside the gray value histogram,

the vertical green line represents the threshold, which is used by the software for identifying the particles on the cross-section of the specimen. After calibration, the microscope scans a user-defined area of the sample, whereas particles with a gray value smaller than the defined threshold and larger than 1 μm are considered. With current SEM settings the automated analysis is performed with a magnification of 400x. The resolution of the images is 1 μm per 3 pixel and therefore the smallest measured particle size is 9 pixels. [9]

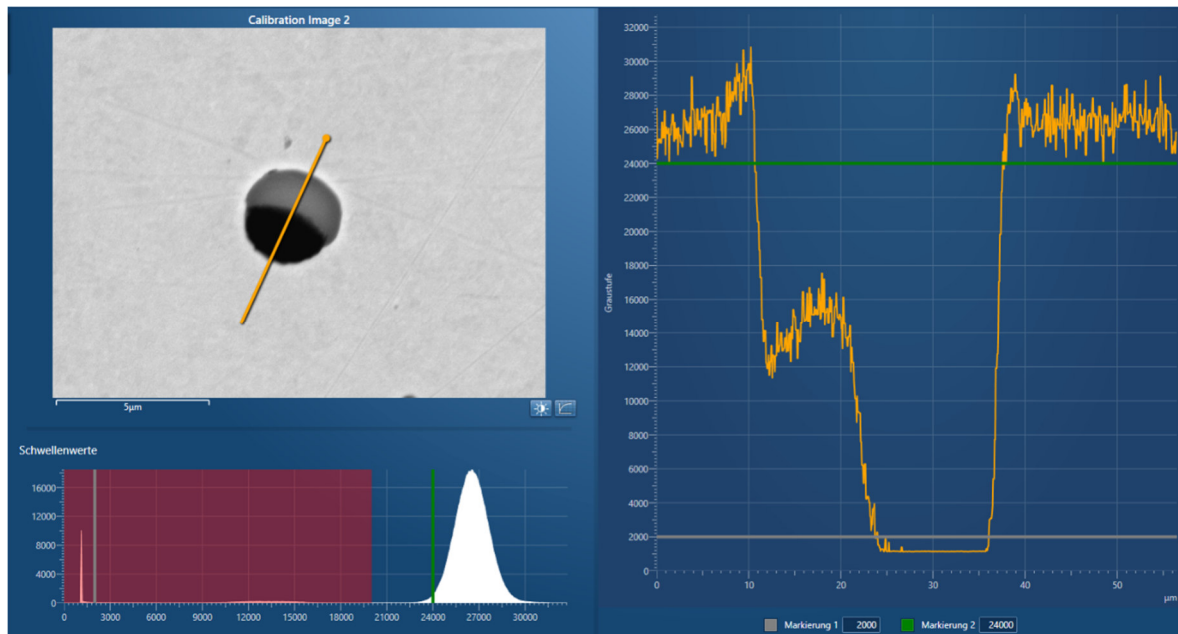


Figure 2-3: Gray value calibration of a multi-phase NMI for automated SEM/EDS analysis

During automated analysis, the chemical composition of identified particles is measured for 1 second with an EDS detector. The software calculates geometric parameters of NMIs such as length, breadth, area, perimeter, equivalent circle diameter (ECD), shape, aspect ratio and mean gray level. Furthermore, the backscattered electron images are stored inside the software.

2.3.2 Feature Evaluation Tool

The Feature Evaluation Tool (FET), developed by Mayerhofer [9] at the Chair of Ferrous Metallurgy, is a stand-alone Matlab program for objective evaluation of automated SEM/EDS measurement data. This tool contains different modules dealing with artefact correction, classification of NMI based on the non-metallic phase, typification based on metallic binding partners without rating as well as data interpretation and representation. The input data for the FET is the as an excel sheet exported information of particles from the SEM software.

In the feature evaluation tool automated SEM/EDS data is processed as follows [9]:

- **Artifact correction:** For micro cleanliness all inclusions larger than 15 μm ECD are excluded from further evaluations. Furthermore, if any particle fulfills at least one of the in **Table 2-I** mentioned criteria, then it will be removed from the actual inclusion excel sheet and moved to a rejected excel sheet.

Table 2-I: Artifact correction criteria

Artefact type	Criterion
Matrix	$\text{Fe} + \text{Mn} + \text{C} + \text{Cr} + \text{Ni} + \text{Mo} + \text{Nb} + \text{Ti} + \text{V} > 99.99$
Grinding residues	$\text{Fe} + \text{Mn} + \text{C} + \text{Cr} + \text{Ni} + \text{Mo} + \text{Nb} + \text{Ti} + \text{V} + \text{Si} > 99.99$
Polishing residues	$\text{Fe} + \text{Mn} + \text{C} + \text{Cr} + \text{Ni} + \text{Mo} + \text{Nb} + \text{Ti} + \text{V} + \text{Si} + \text{Alkali} > 99.99$ Alkali = Ar, Cl, F, Na, K, Cs, P
Insufficient measurement	$\text{O} + \text{N} + \text{S} = 0$

- **Classification:** Depending on present non-metallic bonding partners in NMIs, classification is done in single- (O, N, S) and multi-phase (ON, OS, NS, ONS) inclusions. **Table 2-II** shows the weight percentage thresholds of the classes. To define classes even more precisely, multi-phase inclusions are quantified based on the share of non-metallic phases in 25 %, 50 % or 75 % categories. Symbols '<' and '>' between O, N and S, are used as a representation for the categories. **Figure 2-4** displays possible combinations.

Table 2-II: Classification thresholds

Class	Oxygen [wt. %]	Sulfur [wt. %]	Nitrogen [wt. %]
Oxide (O)	> 0.1	≤ 0.1	≤ 0.1
Nitride (N)	≤ 0.1	≤ 0.1	> 0.1
Sulfide (S)	≤ 0.1	> 0.1	≤ 0.1
Oxide-Nitride (ON)	> 0.1	≤ 0.1	> 0.1
Oxide-Sulfide (OS)	> 0.1	> 0.1	≤ 0.1
Nitride-Sulfide (NS)	≤ 0.1	> 0.1	> 0.1
Oxide-Nitride-Sulfide (ONS)	> 0.1	> 0.1	> 0.1

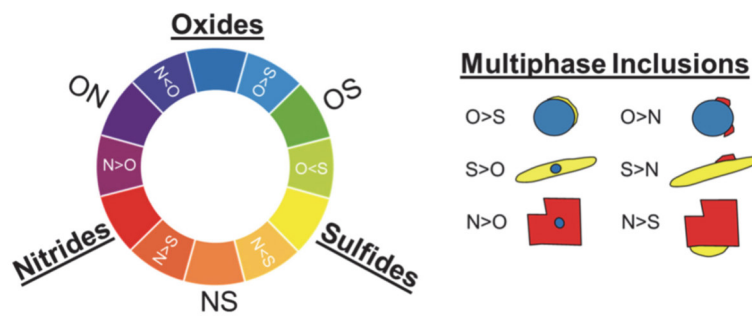


Figure 2-4: Further classification of multi-phase inclusions based on the share of the non-metallic phase [9]

- Typification: Depending on present metallic bonding partners in NMIs, a typification is performed to define inclusions in one of 577 different types. A detailed description is given in [9].

2.3.3 Backscattered Electron Images

Aside from geometric parameters and chemical composition of NMIs, which enable the classification with the FET, BSE images are recorded and saved inside the SEM software during automated SEM/EDS measurement. The gray value of these images relates to the atoms present in the information area of the electron beam. Heavier elements, for example Fe in the matrix, lead to higher gray values, whereas lighter elements such as Al, Mg, etc., which are present in the non-metallic inclusions, result in lower gray values. Automated SEM/EDS measurements can be performed by setting a gray value threshold between the matrix and particles, as non-metallic inclusions generally appear darker than the surrounding matrix in backscattered electron images. [25]

3 Data Science Methods for Inclusion Characterization

The manufacturing industry is facing new potentials and challenges due to the unprecedented increase in available data, often referred to as Big Data. Sustainable improvement in process and product quality can be achieved through the availability of quality-related data. Misinterpreting and mishandling information may lead to distraction and to wrong conclusions about actions. The field of machine learning enables the ability for manufacturers to utilize the vast amounts of data. Machine learning (ML) methods, such as Support Vector Machine (SVM), are specifically designed to effectively analyze large amounts of data with high dimensionality. [1]

3.1 Principal Component Analysis on Inclusion Datasets

Principal Component Analysis (PCA) is a data science technique used for dimensionality reduction by extracting essential information from multivariate datasets and representing it as a set of new orthogonal variables referred to as principal components (PC). The original variables, for example the chemical composition of an NMI, exhibit a linear relationship with the new principal components. PC1, the first principal component, is calculated by finding the linear combination of the original variables that describes the maximum variance in the dataset. The second PC represents the second greatest variance, PC3 the third greatest and so on. In most cases the first two PCs represent 80 % of the variance, which enables the plotting of multivariate datasets on two-dimensional scatter plots with minimal information loss. [26]

A study by Abdulsalam et al. [26] showed an application example of PCA on automated SEM/EDS generated NMI datasets. The dimensionality reduction reduced the number of variables, in this case the Al-, Mg-, Si-, Mn-, Ca- and S- content of a 4140 steel (42CrMo4), to two principal components PC1 and PC2. Both PCs combined retained 95 % of the original dataset's variance and showed following relationship with the original variables:

$$\begin{aligned} PC1 &= -0.027Mg - 0.716Al + 0.002Si + 0.468S - 0.198Ca - 0.007Ti + 0.478Mn \\ PC2 &= 0.007Mg + 0.331Al - 0.008Si - 0.391S - 0.569Ca - 0.014Ti + 0.643Mn \end{aligned} \quad (\text{Eq. 1})$$

The two remaining principal components enabled a visualization of the entire inclusion population in a single plot and therefore a fast sample evaluation can be carried out by metallurgists during process control. **Figure 3-1** shows the result of the PCA on four different 4140 steel samples. Among sample A and B, inclusion populations were very similar. The same was observed for sample C and D. Comparing sample A and B with C and D, inclusion chemistry differed significantly due to Al and Ca containing inclusions at low PC1 values.

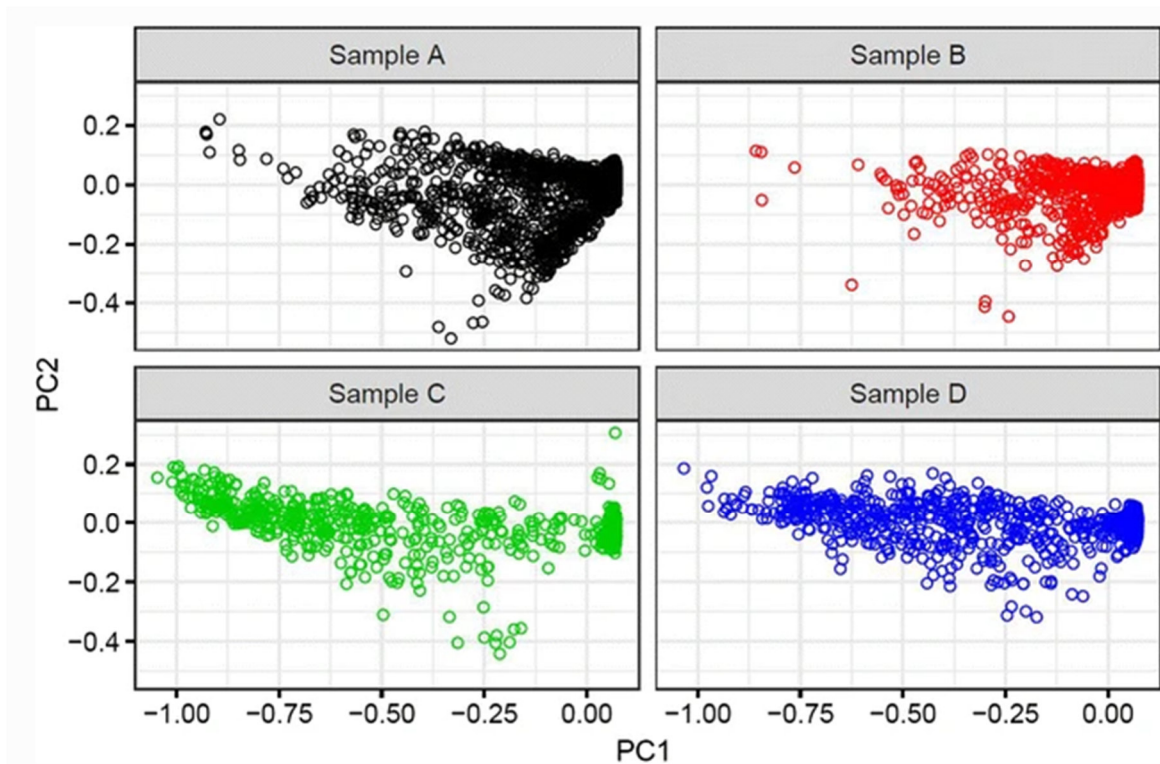


Figure 3-1: PCA scatter plots of four samples from a 4140 steel [26]

3.2 Machine Learning for Determining the Abundance of Inclusions

The abundance of inclusions can be detrimental to the quality and performance of a material or product which causes increasing costs and further post-processing manufacturing steps during production. Early detection of the inclusion content in the steelmaking process, along with appropriate countermeasures, can effectively address this problem. A data analysis and machine learning approach to develop a decision-support tool helping to minimize the inclusion content was developed by Mesa et al. [6]. In this publication, the abundance of inclusions in austenitic steel was linked to the first stage of the manufacturing process using more than 300 variables from the melting shop stage. The variables described the conditions during tapping,

steelmaking (temperature, time of treatment, amount and type of additions) and casting. Slab dimensions and defect classification in laboratory tests were also included. Various machine learning models such as linear regression, random forest, artificial neural networks, and support vector machines were fitted to the data. **Figure 3-2** shows a scatter plot representation comparing the observed and predicted values of the average abundance of inclusions. The red data points correspond to predicted values outside the 95 % confidence interval.

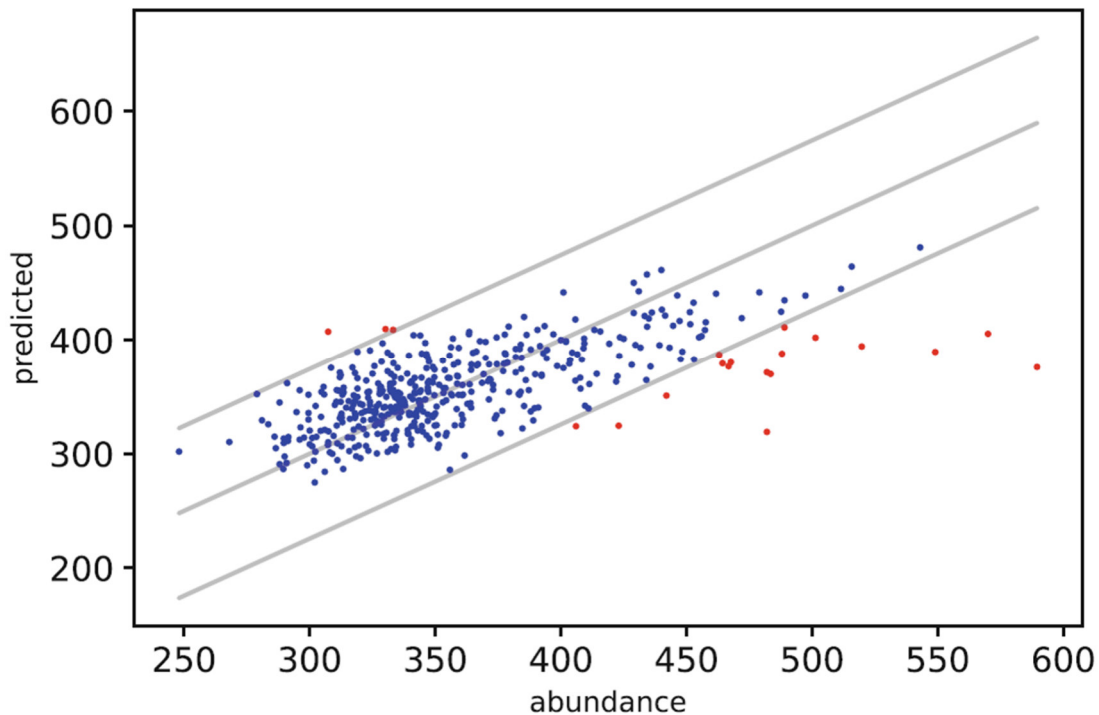


Figure 3-2: Comparison of observed and predicted values of the average abundance of inclusions from a linear regression model [6]

3.3 Classification of Non-metallic Inclusions

In contrast to automated SEM/EDS measurements, using machine learning tools for classifying non-metallic inclusions is a rapid and low energy consuming method of determining steel cleanliness. Regarding the training data of the machine learning algorithms, two different approaches can be considered. Ramesh Babu et al. [27] and Abdulsalam et al. [28] used geometric output data of automated SEM/EDS analysis to perform inclusion characterization and cluster detection. BSE-images can also fulfill the purpose of serving as input data for machine learning. Previous works [29] on this field showed the application of convolutional neural networks for predicting inclusions classes by using BSE-images. **Figure 3-3** gives a

schematic overview of the possible NMI classification approaches with data from the Aztec SEM software. The morphology/geometric data as well as BSE-images are recorded during automated SEM/EDS analysis and represent the input data for different ML techniques.

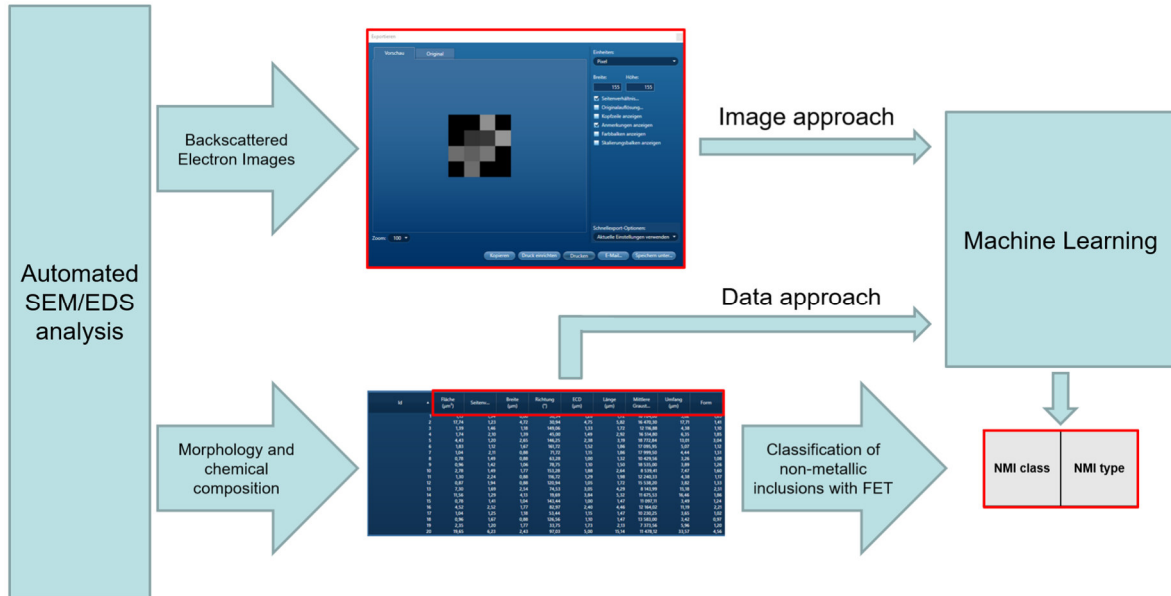


Figure 3-3: Schematic overview of different data approaches for the application of machine learning for inclusion characterization

3.3.1 Geometric Data Based Classification

Geometric data of NMIs contain information about the size, shape, and gray value properties, which can be used as training data for machine learning techniques. Several classification algorithms, including Naïve Bayes, SVM, and Random Forest (RF), have been tested for their effectiveness in performing inclusion classification tasks [26,27]. These algorithms have demonstrated their ability to achieve a certain level of accuracy. As published by Babu et al. [27], using SVM for the binary inclusion/rejected classification was possible with an accuracy of 89 %. For 8-class classification, including single- and multi-phase NMIs, the accuracy dropped to 61 %. The achieved classification performance is a function of various parameters. Contrast and brightness settings of the microscope influence BSE-image quality and the calculated geometric data such as the mean gray value. The improvement of SEM settings for inclusion classification by Ramesh Babu et al. [27] included maximizing the brightness and contrast difference between the oxide and sulfide part within an oxide-sulfide inclusion. The result of this change can be seen in **Figure 3-4**. Lower contrast led to similar gray values for oxide and sulfide in oxide-sulfide inclusions during automated SEM/EDS measurement.

Increasing the contrast resulted in a more pronounced difference in gray values and improved the accuracy of binary classification to 98 % and 8-class classification to 81 %.

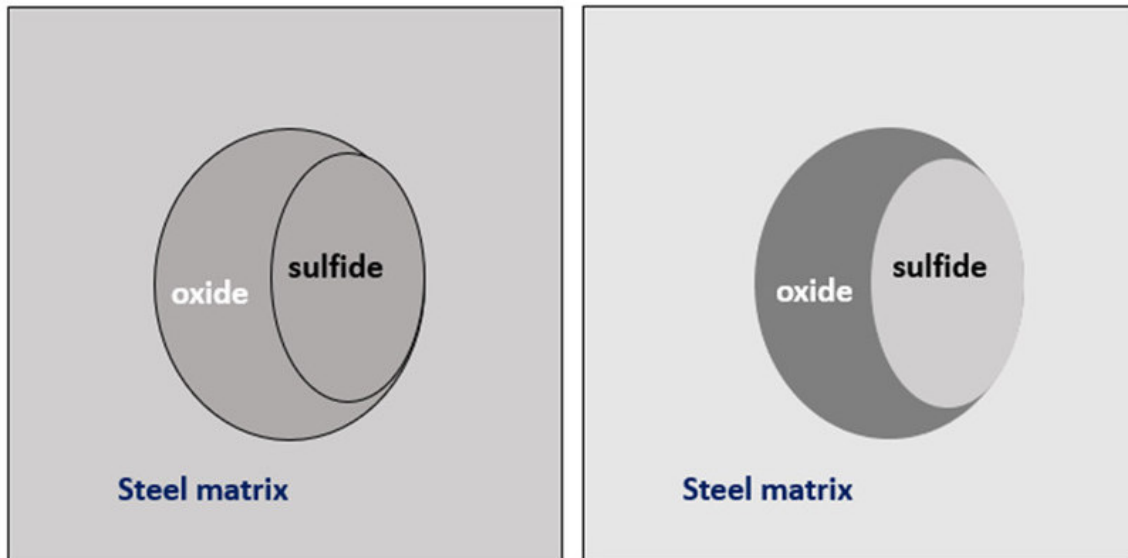


Figure 3-4: Influence of contrast on the gray value of oxide and sulfide in oxide-sulfide inclusions [27]

Other metallurgical influencing factors that affect classification performance include the type of steel and its corresponding inclusion population, the complexity of inclusions, the number of inclusion classes defined by the metallurgist, and the evaluation method used to define the element-thresholds for inclusion classes.

3.3.2 Image Data Based Classification

Abdulsalam et al. [30] as well as previous work [29] conducted at the Chair of Ferrous Metallurgy demonstrated the successful application of machine learning techniques for NMI classification using BSE-images. The convolutional neural network (CNN) is a well-known deep learning model for image classification and was utilized in both studies for training and testing. Convolutional layers are especially useful as they can extract important features such as edges [31]. CNNs process images directly and require data preprocessing if variations in image sizes occur. In previous work [29], images were exported from the Aztec SEM software with a size of 320x313. The VGG16 architecture, which is a CNN, uses 224x224 sized images with three channels in its input layer. Therefore, image scaling operations were necessary to enable the classification with this architecture. After training the VGG16 on 17300 images, a validation accuracy of 71,7 % could be achieved within the 4-class classification (OS, Oxide, Sulfide, Rejected). Data complexity and hyperparameter of the CNN showed a significant

impact on the achievable accuracy. After optimizing model parameter and data cleaning steps, which simplified the NMI classes, the accuracy could be increased to 80 %. Main problems during classification occurred for the Oxide/Rejected classes, as it is shown in the confusion matrix from the test dataset in **Figure 3-5**. The second highest error happened during OS/Rejected classification.

True class (determined by the FET)	OS	381	7	74	20
	Oxide	64	99	132	45
	Rejected	18	18	208	23
	Sulfide	9	4	47	480
		OS	Oxide	Rejected	Sulfide
		Predicted class (determined by CNN)			

Figure 3-5: Confusion matrix of an VGG16 network trained on BSE-images [29]

The reliability of EDS measurements needs to be considered for the application of ML techniques. Standardized classification criteria determine the inclusion class, which is subsequent the ground truth label for the algorithm. Due to a drift in BSE imaging, the measured composition can differ from its actual one. This problem can lead to wrong determined image labels, as it is shown in **Figure 3-6** for the case of mislabeling multiphase inclusion as single phase NMIs. The measured chemical composition, represented by the bars, would lead to the conclusion that only one phase is present. However, the BSE images clearly show that the NMIs consists of two phases. Performance of machine learning algorithms relies on the quality of training data and is directly influenced by the reliability of EDS measurement. Mislabeling NMIs increases the number of errors during classification. Furthermore, CNN models with a large amount of optimizable parameters require a high computational cost during training. Another drawback of deep learning algorithms is that the basis for classification can be difficult or impossible to extract. Using Random Forest classifier for predicting inclusion classes nullifies this problem because information about the most important features of the input data for prediction is accessible. RF methods are not as well suited for image classification tasks as CNNs, but due to lesser amount of parameters, the computational cost

is lower. If the tradeoff of the accuracy using RF classifier is not significant, then it may be a suitable choice for classifying BSE images. [30]

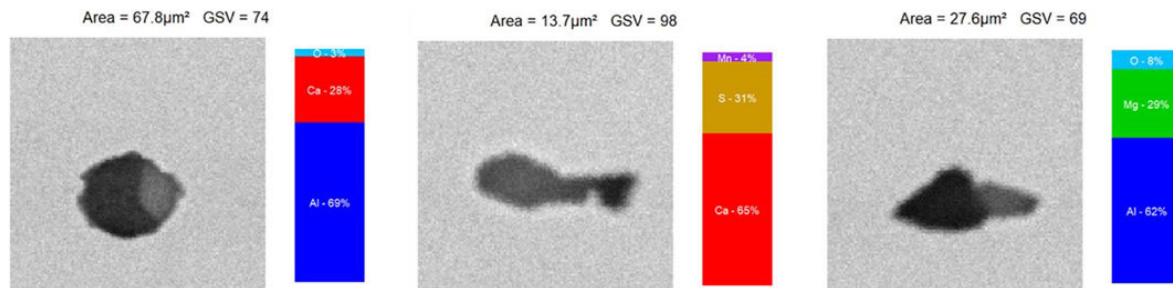


Figure 3-6: Multiphase inclusions with wrong determined chemical composition [30]

4 Data Preprocessing

In this work, the suitability of various machine learning methods for application as an inclusion characterization tool was tested and evaluated. Training and test datasets were generated by automated SEM/EDS measurements in combination with data preprocessing techniques. In order to use BSE images from Aztec software as input data for machine learning, a new method of preprocessing had to be devised. The integrated development environment Spyder with Python version 3.9.15 was utilized to satisfy the requirements on the quality of the input data for machine learning.

4.1 Data Pipeline for Exporting SEM/EDS Data

At the Chair of Ferrous Metallurgy, inclusion characterization is mainly done by automated SEM/EDS analysis combined with a post-measurement evaluation using the Feature

Evaluation Tool. During this process, an excel list containing classified and typified NMIs with calculated geometric parameters and determined weight percentages of measured elements is generated. To enhance knowledge and gain information about complex inclusions, manual mappings and high-resolution BSE images can be recorded. In contrast, BSE images for each analyzed particle during automated SEM/EDS measurement are low-resolution and generally not used for any other application besides the newly developed inclusion characterization approach with machine learning. A new technique for extracting this image data from the Aztec environment had to be developed, since Aztec version 5.0 did not provide a built-in function for exporting every image from the software. Overview of the whole data pipeline for dealing with automated SEM/EDS data shows the flowchart in **Figure 4-1**. Colors of the process steps refer symbolic to different program environments, in which the user or script is processing the data. Starting with automated SEM/EDS measurement in Aztec software, information about different geometric properties and chemical composition is measured and calculated as well as BSE images are recorded and saved during analysis of specimen. The conventional way of NMI characterization consists of copying geometric and chemical data from Aztec software in an excel sheet, which is stored in a folder inside the windows environment. For a further detailed evaluation, such as determining inclusion class and type, the data in the excel sheet is processed by the FET. During this process, a 'Typified Excel Sheet' listing all the defined non-metallic inclusions and a 'Rejected Excel Sheet' containing the data which gets rejected based on defined rules are generated. For the combination of several typified and rejected files from automated SEM/EDS measurements of different steels, a python script 'Generate Excel Database' puts geometric and chemical information of NMIs together in one excel database, which is necessary for preprocessing the data for machine learning. In the 'Excel ML Database', an automatic defined identification number as well as corresponding file- and project name gets assigned to each NMI for ensuring traceability after combination. Furthermore, information of this database is used during image renaming and class sorting as it provides the link between project specific nomenclature (ID inside Aztec software; starts with 1 for every new measurement) and database specific nomenclature (progressive ID; every NMI gets a unique number).

A python-programmed macro is used for semi-automatic exportation of BSE images from Aztec software. After setting up mouse positions, movement, and functions, the macro repeats the defined steps for a certain number of times, depending on how much NMIs needs to get exported. During this process, image files get saved in one folder. As already described, the python script 'Image Renaming and Class-Sorting' accesses the necessary information from the previously generated database. After renaming images to their progressive ID and sorting them into folders of their respective class, image processing is done by the python script 'Aztec

Image Processing'. This step includes image padding to a size of 60x60 and image reshaping to a one-dimensional vector. Padding, which is filling up missing pixel values with zeros to the desired size, ensures, that information about the original size doesn't get lost because the reshaping process is revertible with a predefined image size. Due to the fact that this process artificially enhances the number of pixels with zeros in BSE images, this gray value will not be included in the evaluation. Excluding zeros from evaluation doesn't lead to an information loss of inclusion data because already in Aztec software, missing pixel values to a rectangular image size are represented by zeros, and no inclusion class contains gray values with this value. The final 'Excel ML Database – Images' contains the ID, class, and processed images and serves as training data for machine learning in the case of direct training with pixel values.

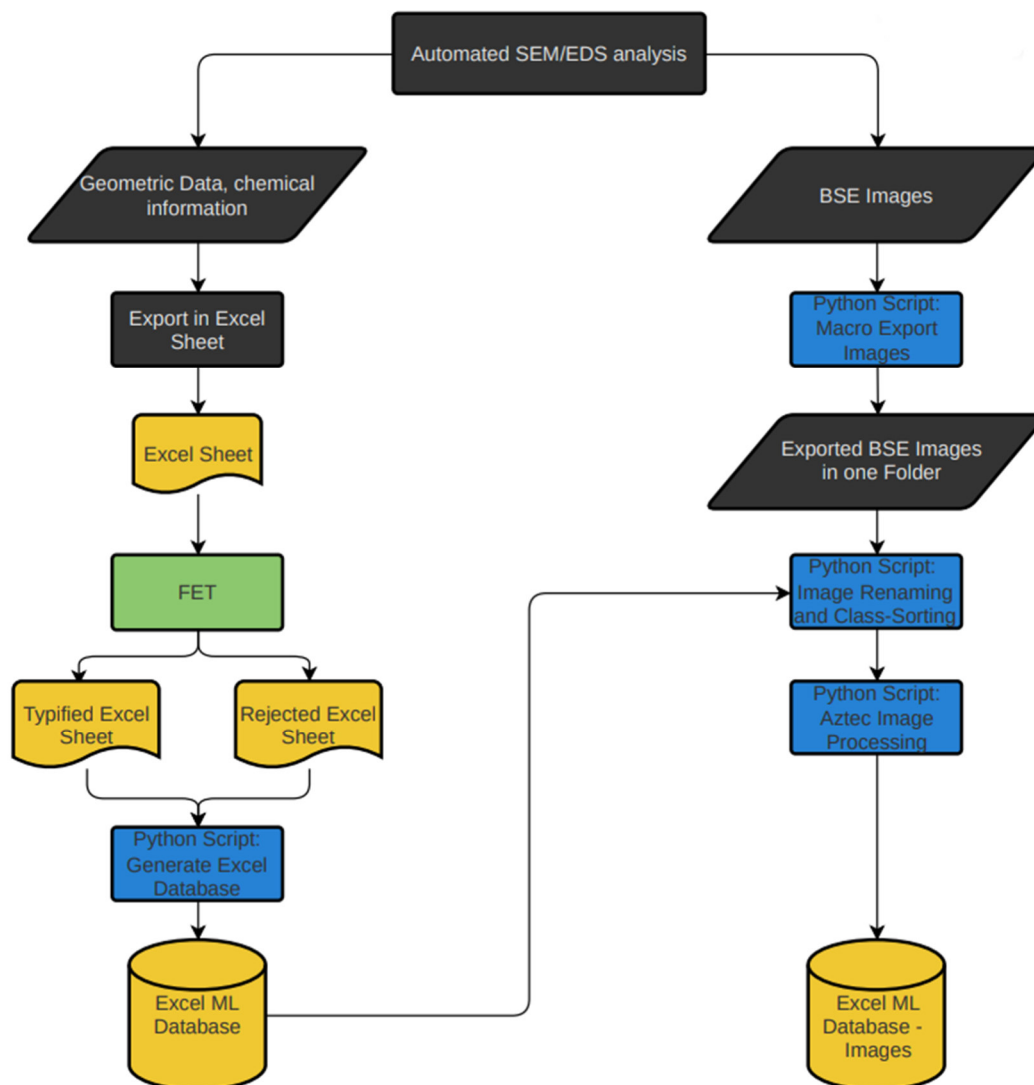


Figure 4-1: Flowchart of how data, starting from Aztec software, is processed (color program environments - black: Aztec, green: FET, yellow: Windows, blue: Python)

4.2 Description and Statistical Overview of the Dataset

The Standard Cross-Industry for Process Data Mining (CRISP-DM) developed a comprehensive process model for carrying out data mining projects, independent on industry sector or used technology [6,32]. The second stage in this methodology, known as ‘Data Understanding’, was performed to gain knowledge about the behavior, relationship, and differentiating factors of NMIs in the generated dataset. Furthermore, statistic observations are important for understanding the relevance of features, derivation of the required preprocessing steps, and ultimately for ensuring high data quality. A feature is an individual measurable property of the process being observed [33]. Any machine learning algorithm can classify data by utilizing a group of features. In this chapter, the dataset is presented along with information on its class and type distributions, and a brief overview of BSE image characteristics is provided.

Seven different steel samples were pressed, cut, grinded, polished, and subsequently observed by means of automated SEM/EDS measurement at the Chair of Ferrous Metallurgy to ensure a large enough dataset with a high NMI class variety. The chemical composition of the steels, obtained with a spark spectrometer, is given in the appendix. **Table 4-I** summarizes the microscope parameters, which remained the same across the analyzed specimen. Contrast and brightness settings were adjusted accordingly for every steel during calibration. Matrix gray value peaked at approximately 26000 with a mean oxide gray value in the range between 1500 and 2000.

Table 4-I: Measurement parameter for field-emitter-based JEOL 7200F

Parameter	Value
Beam energy	15 kV
Probe current	12 pC
Working distance	10 mm
Resolution	1024x960 px
Minimal particle size	9 px
Magnification	400x
EDS measurement time per particle	3 s

The generated SEM/EDS data was exported via the described data pipeline in the previous chapter. The final ‘Excel ML Database’, stored as a csv file, had 38281 entries with 92 columns

containing the ID, originating file- and project name, determined class and type, morphological information, and chemical composition. In addition, the 'Excel ML Database – Images' consists of 38281 images, represented by a one-dimensional vector for each measured particle.

4.2.1 Class Distribution

The initial reason for measuring different steels and combining them into one dataset is to create a balanced class distribution, because depending on production route, metallurgical treatments, steel grade, etc., some NMI classes can dominate the inclusion population. This chapter represents the beginning of the verification process, to determine whether inclusions from different steels can actually be combined or differ too much so that every steel, even though the class distribution might be imbalanced, has to be treated separately for machine learning. **Figure 4-2** shows the absolute number of particles of each class within a steel grade in a stacked bar plot with the measured sample area on top of each bar. The class distributions varied between the different steels. The rail and construction steel were dominated by sulfides, whereas the spring and bearing steel mostly contained OS. The micro alloyed and quenched and tempered (QT) steel were evenly distributed regarding the inclusion classes oxide, sulfide, and OS, but showed compared to the other steels a high amount of rejected data. The overall number of NMIs in the austenitic steel was significantly lower compared to the other samples.

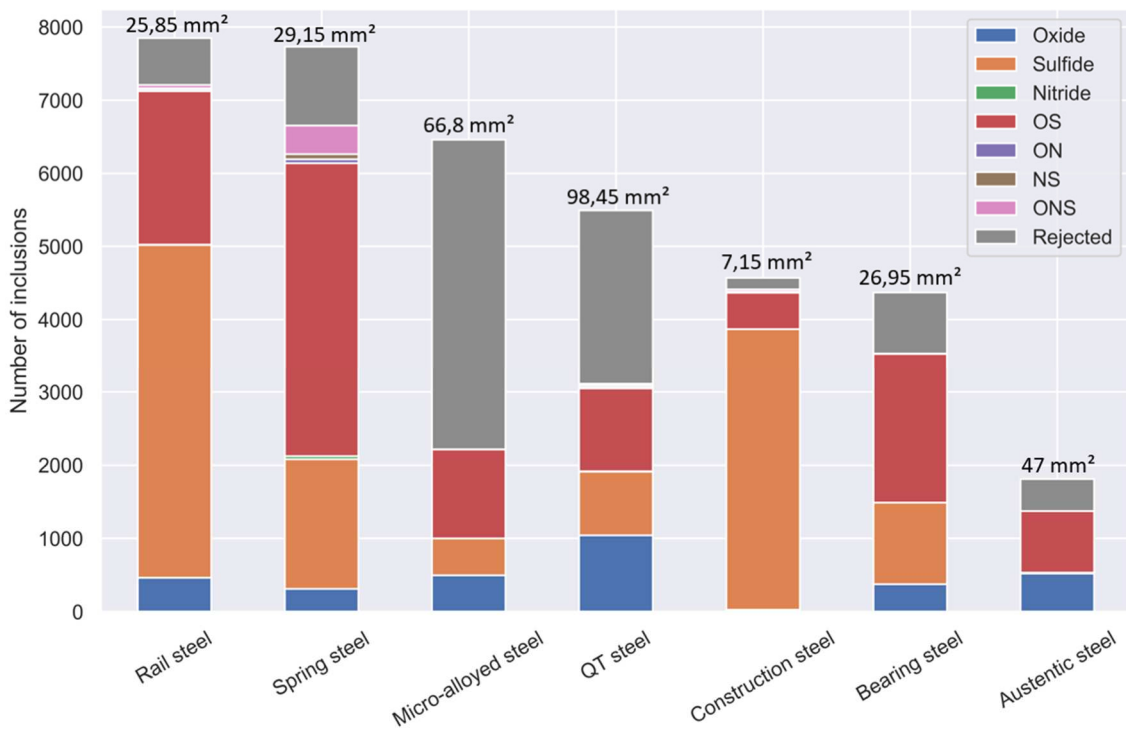


Figure 4-2: Number of NMIs in each class for the different steel samples

Relating the absolute amount of NMIs to the measured sample area leads to the bar plot in **Figure 4-3**. The construction steel was well above the other steel grades. Rail and spring steel showed a similar number of inclusions per mm² measured area. Micro alloyed, QT, and austenitic steel had a low inclusion quantity.

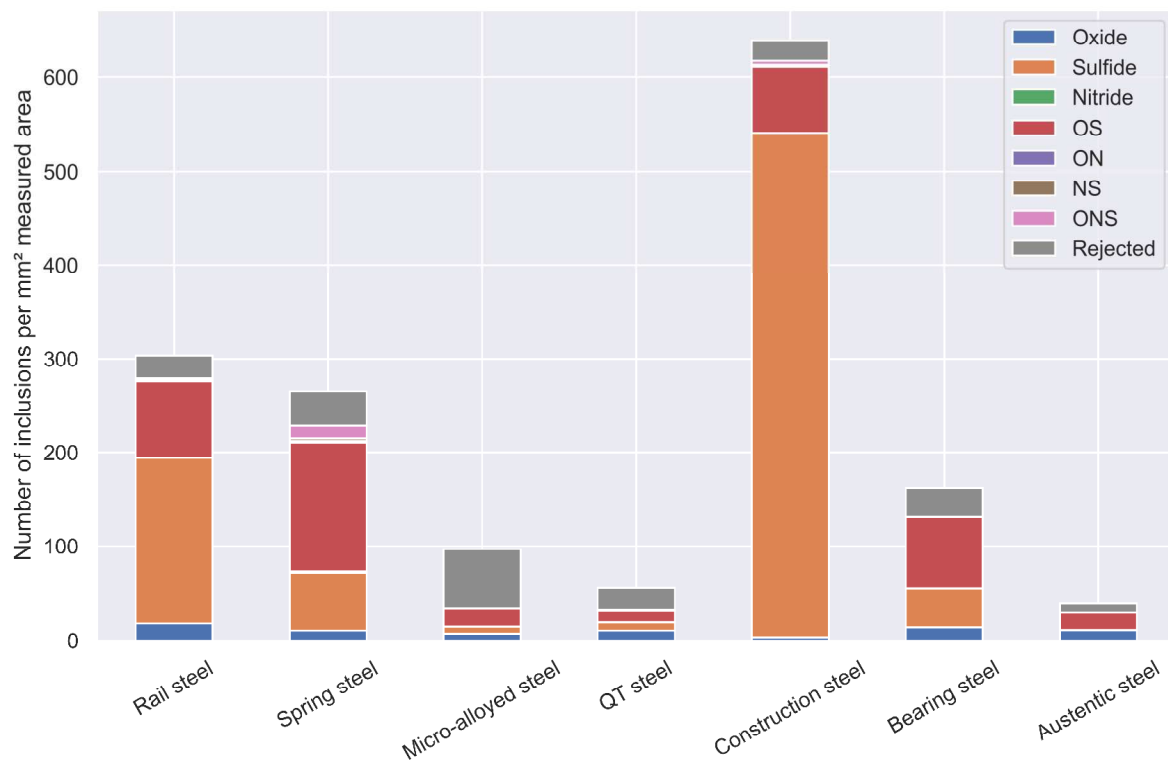


Figure 4-3: Number of NMIs per mm² in each class for the different steel samples

More important for machine learning is the overall distribution of the NMI classes, if class labeling is used in the input data. In this case, the number of inclusions assigned to a specific class define the actual balance of the data. An overall overview of the inclusion class distribution inside the whole dataset is given **Figure 4-4**. Sulfides, OS, and rejected data were the three dominating classes, representing 88 % of the inclusion population. The number of oxides was significantly lower. The remaining classes, such as ONS, NS, ON, and nitrides, barely existed in this dataset. Due to this pronounced imbalance, classes apart from sulfide, OS, rejected, and oxide, which represent only 2 % of the data, were removed from the dataset for machine learning.

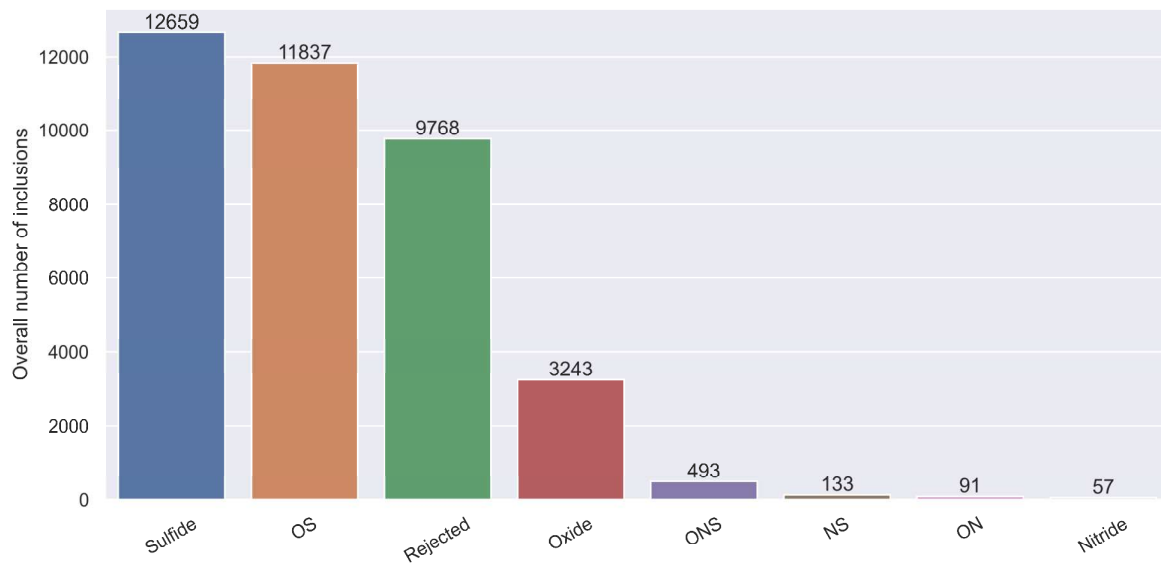


Figure 4-4: Number of NMIs in each class for the whole dataset

4.2.2 Type Distribution

For a more advanced insight into the behavior of NMI classes, inclusion types need to be considered because class labels may have too much variation to show a recognizable and unique pattern for machine learning algorithm. The Feature Evaluation Tool defines, aside the more general describing class terms, very specific type labels depending on the measured chemical composition of the respective particle. The variation of different types is exemplary shown in **Figure 4-5** for oxide inclusions in the QT steel by plotting the weight percentage of oxygen against the mean gray level. MA-spinel, type $(Mg,Al)O$, showed a cluster at low mean gray level with higher oxygen content. Other types, mainly Ca-, Si-, Al/Si-, and Al-oxides, had lower amount of oxygen and were located at higher mean gray level. This illustrated complexity needed to be considered if the mean gray level was used as an input parameter for machine learning to describe inclusions in the oxide class. Furthermore, this pronounced oxide type variation may not be as significant in other steels or in the case of other inclusion classes. For a detailed description of the class complexity regarding BSE images, refer to Chapter 4.3.3.

A high intra-class variation leads to problems for classification with machine learning models, as the performance can drop sharply when mismatch between training and testing data distributions occurs [34,35]. Automated SEM/EDS measurements, or more general parameters affecting the output of NMI inclusion analysis, may be susceptible to test domain shifts, which are frequently encountered in real-world applications. Mayerhofer [9] defines nine

influencing factors, such as raw materials, melt, furnace, solidification, cooling process, sampling, measurement, evaluation and final data representation. The problem with test domain shifts for inclusion characterization is only being mentioned and will not be further addressed in this work. However, the observed complexity of inclusion classes led to the consideration of a splitting process using type specific labeling for classification.

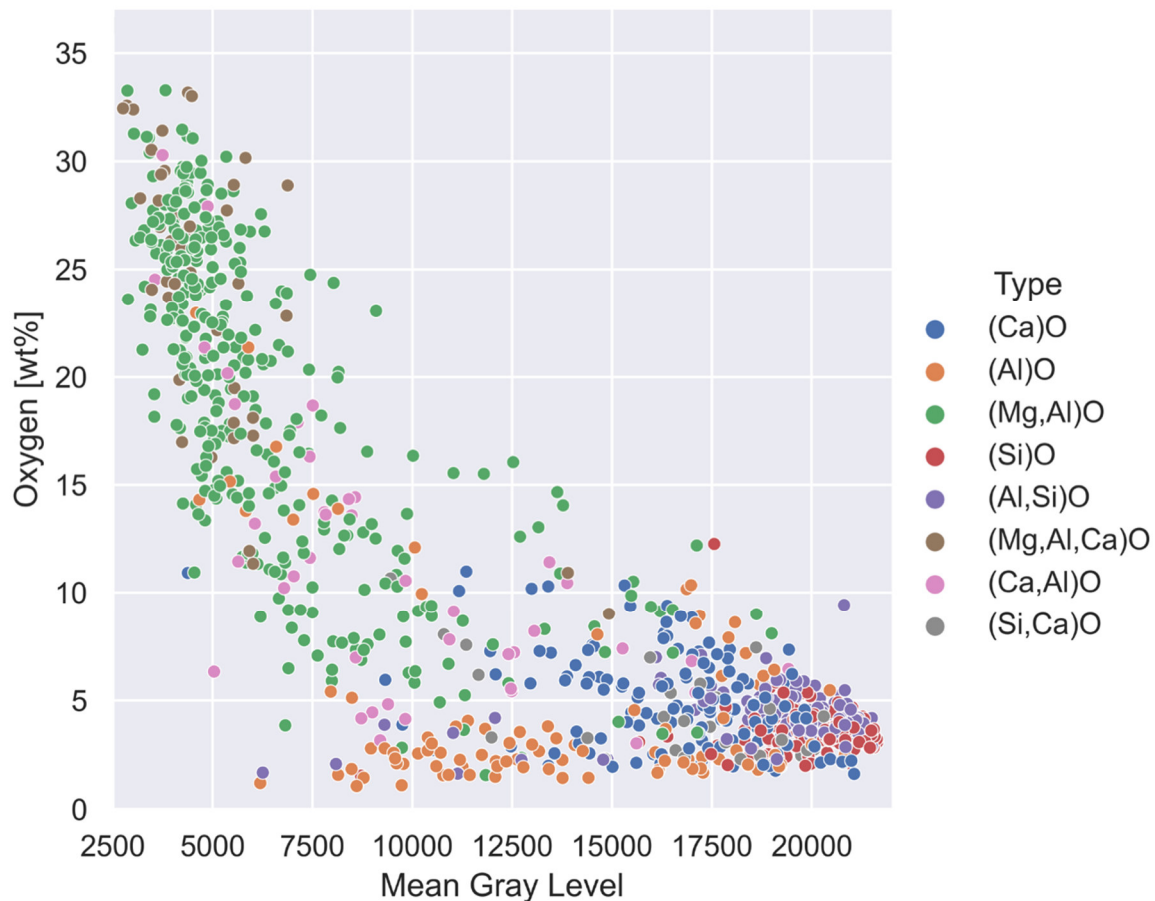


Figure 4-5: Oxide inclusion types in the QT steel

Overall, 286 types were present across all steel grades. **Figure 4-6** shows the amount of the twenty most common types in the dataset. MnS and (Mn,Ca)S as well as rejected particles (not typified, SiC/matrix) represented almost 50 % off all NMs. The other half contained 282 types, which increased the variety tremendously. Following definition was used in the Feature Evaluation Tool regarding the most important rejected types [9]:

- Not typified: Particle with O, N, and/or S content, but no metallic bonding partner
- SiC/matrix (artifact type – grinding residues): $\text{Fe} + \text{Mn} + \text{C} + \text{Cr} + \text{Ni} + \text{Mo} + \text{Nb} + \text{Ti} + \text{V} + \text{Si} > 99.99 \%$
- Matrix: $\text{Fe} + \text{Mn} + \text{C} + \text{Cr} + \text{Ni} + \text{Mo} + \text{Nb} + \text{Ti} + \text{V} > 99.99 \%$

A relatively even distribution was present among the different types of particles, with the exception of the two dominant ones, namely '(Mn)S', and 'not typified'. This resulted in a challenge for the definition of the used labels for machine learning, as introducing additional types led to a more difficult task. Furthermore, not all types showed sufficient amount of data to enable the detection of patterns for machine learning algorithms. 1000 images per class is a rule of thumb for the minimum number in the training set for computer vision applications [36]. Applying this to the distribution in **Figure 4-6**, nine types could be considered for a type specific labeling. For a comparison of the classification performance, NMI types and classes were used in separate training sets as labels.

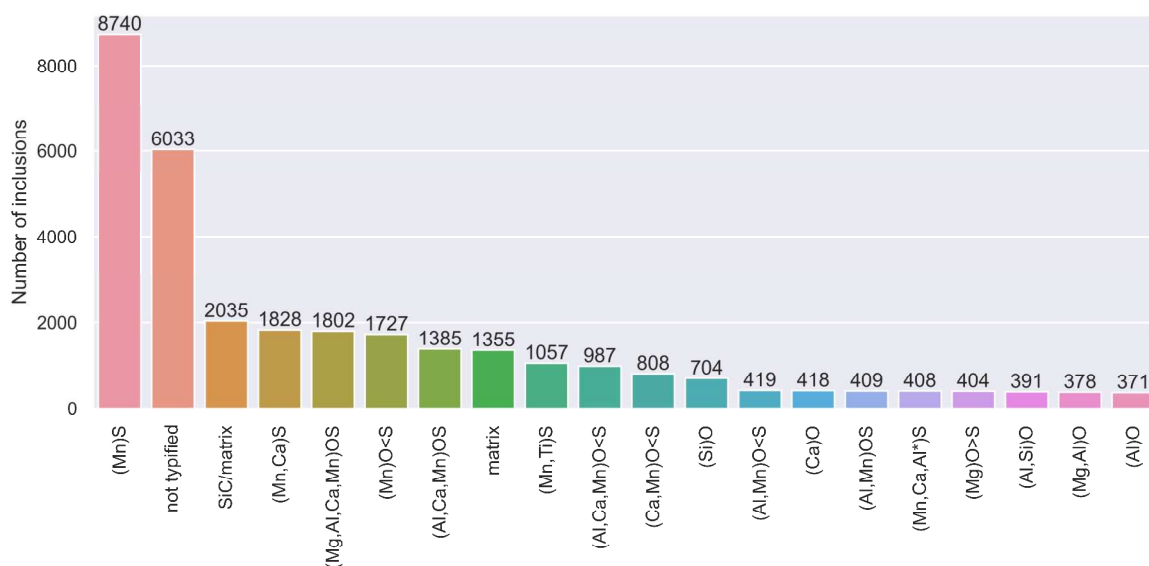


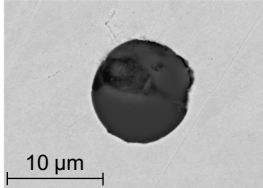
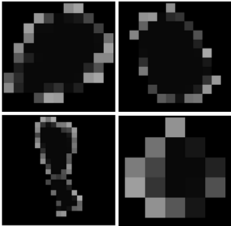
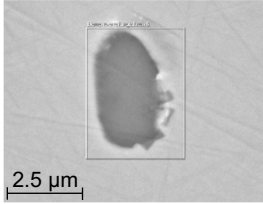
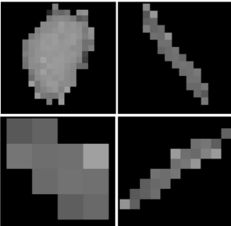
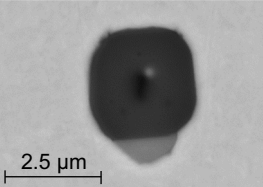
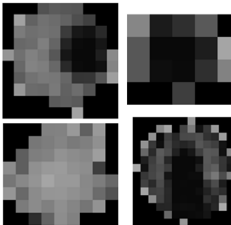
Figure 4-6: Amount of the twenty most common types in the dataset

4.2.3 BSE Images

As already discussed, gray value in BSE images depends on present elements in the measured area. Furthermore, different physical properties can lead to specific characteristics of NMI classes. In this section, the appearance of oxide, sulfide, and oxide-sulfide inclusions are compared and discussed. **Table 4-II** shows high- and low-resolution images of these classes. The high-resolution images were recorded manually whereas the low-resolution ones were generated during the automated SEM/EDS measurement. Oxides with metallic bonding partner such as Mg, Al, Si, Ca, Ti, Cr or Mn relate with a low gray value and a dark appearance. Low-resolution BSE images show a specialty for oxides, which will be referred to as 'gray ring', due to the pronounced influence of matrix at the inclusions' edge. A higher amount of matrix inside the interaction volume of electrons leads to an increase in gray level in this area of the

inclusion [9]. In contrast to oxides, sulfides with a metallic bonding partner such as Mn, Mg, Ca or Ti show higher gray values. The appearance of OS depends on the oxygen/sulfur ratio of the inclusion.

Table 4-II: Comparison of oxide, sulfide, and oxide-sulfide inclusions in high- and low-resolution images

NMI class	High-resolution BSE image	Low-resolution BSE image	Characteristics
Oxide			Shape: predominantly round Gray value: in the center low, gray ring due to matrix influence
Sulfide			Shape: round/elongated Gray value: brighter than oxides, uniform over whole area
Heterogenous Oxide-Sulfide (OS)			Shape: round/elongated Gray value: depends on oxygen/sulfur ratio, higher oxygen content → more pronounced dark area, higher sulfur content → more pronounced bright area

The described characteristics in low-resolution BSE images are more pronounced at larger inclusion dimensions. Influence of matrix can lead to a significant increase in the mean gray level for small oxide inclusions in the range of 1 – 2 μm ECD. **Figure 4-7** shows the oxygen content plotted against the mean gray level of oxide inclusions from the QT steel with a colormap representing the ECD. Larger oxides were located at high oxygen content with low mean gray level, whereas small ones tended to cluster at significantly lower gray values. Sulfides are not affected by this phenomenon, as the difference in gray values between the inclusion and the steel matrix is inherently much smaller.

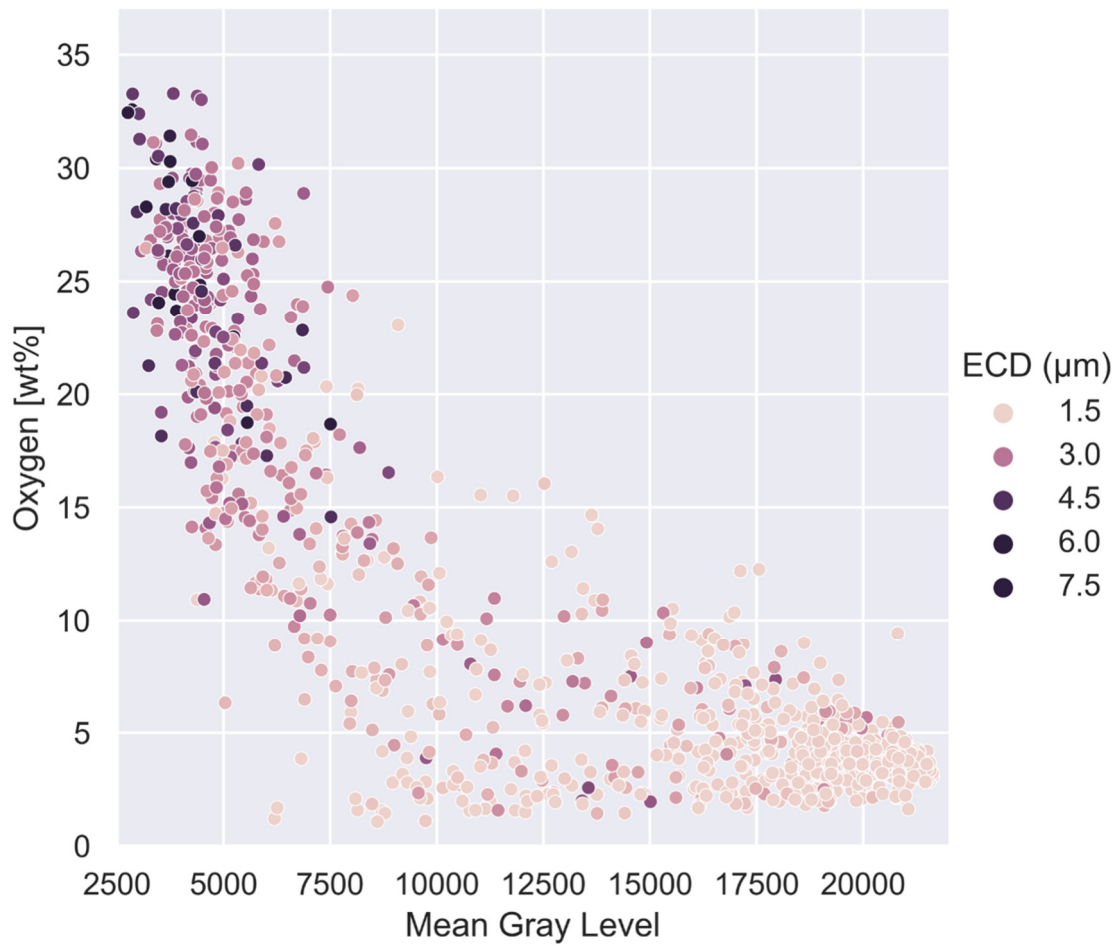


Figure 4-7: Influence of ECD on mean gray level of oxide inclusions in the QT steel

Figure 4-8 shows exemplary images of affected oxide inclusions with small dimensions. Differentiating, if the three images belong to oxides or sulfides is impossible without having access to the measured chemical composition. Small images with low resolution present a problem as they do not provide sufficient information content for machine learning algorithms to extract useful features for characterization. Feature extraction, which is discussed in the following chapter, is difficult in such images.



Figure 4-8: Low-resolution BSE images of small oxide inclusions (ECD \approx 1,5 μ m)

4.3 Training Data

In this work, two different types of training data were used for machine learning. On the one hand, training models directly with pixel values resulted in a low time and preprocessing effort. On the other hand, extracting features from images and using them as training data simplified the classification task, because instead of 3600 pixels from 60x60 sized images only a certain amount of input variables represented the whole dataset. Furthermore, flexibility and adaptability increased due to the ability of defining task-specific features. This chapter deals with the definition of different features inside the dataset, which were subsequently used as input variables for machine learning. Manual and algorithmically extracted features are discussed, compared, and calculated for statistical comparison of NMIs across the different steel grades.

4.3.1 Feature Definition

As mentioned earlier, training machine learning models directly with BSE images resulted in 3600 input variables, which are referred to in this work as pixel features. Aztec software determines during the automated SEM/EDS measurement geometric properties of every measured particle. Theoretically, this information can also be exported manually from BSE images. Geometric properties are referred to as geometric features. Furthermore, new input variables can be defined using information from BSE images.

A grey value histogram is a visual representation of the distribution of pixel intensity values in an image. **Figure 4-9** shows the gray value histogram of the dataset, containing information of every pixel from all BSE images. Contrary to previous calculations, the extracted image data from Aztec software has a color depth of eight bit, which causes the gray value to be in the range between zero and 256. Plotting all values between 0 and 160 in **Figure 4-9** ensures that no information is lost. The large number of histogram classes are simplified by labeling every 25th bar. Two peaks are visible in the histogram, which suggests that there were two dominant intensities present in the BSE images. The first peak at 0 was caused by the padding process inside Aztec software (images must be rectangular) and during image preprocessing (image resizing to 60x60). This peak showed no importance for the analysis of the images as it did not provide meaningful information and was ignored during further evaluations. The second peak at a gray value of 9 indicated a usable feature for classification. Images from NMI classes such as OS, rejected data, and oxides increased this gray value tremendous, whereas the influence of BSE images from sulfides was almost neglectable (**Figure 4-10**).

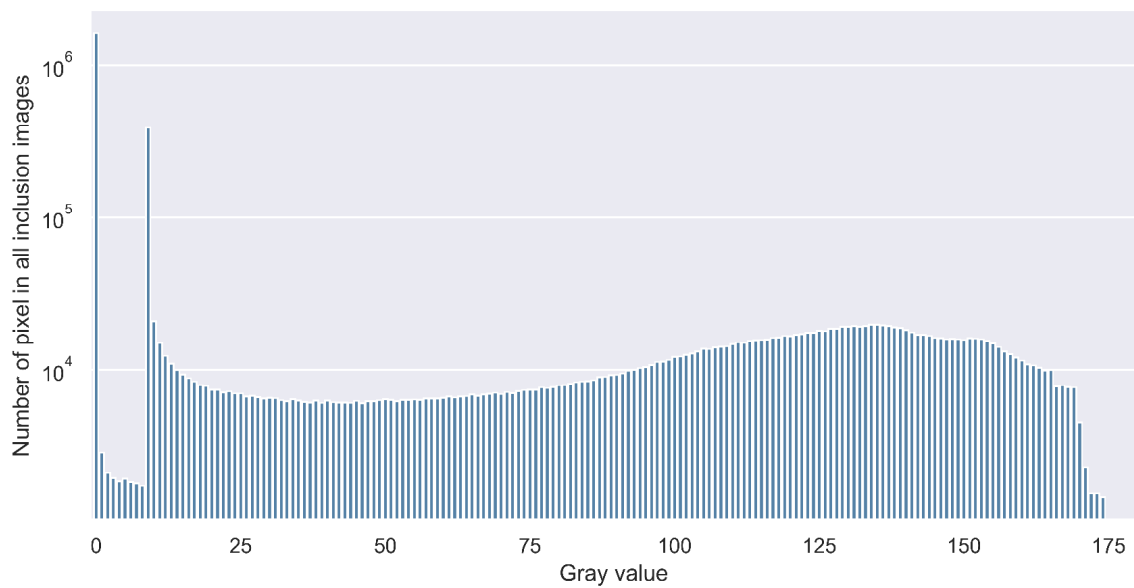


Figure 4-9: Gray value distribution of every image in the dataset

BSE images from sulfides only had 3737 pixels with a gray value of 9. Considering the high amount of 12659 sulfides in the dataset, this represented a share of 0,007 % of all pixels. The number of pixels in BSE images with this gray value indicated an appropriate feature, especially for a sulfide vs. rest classification.

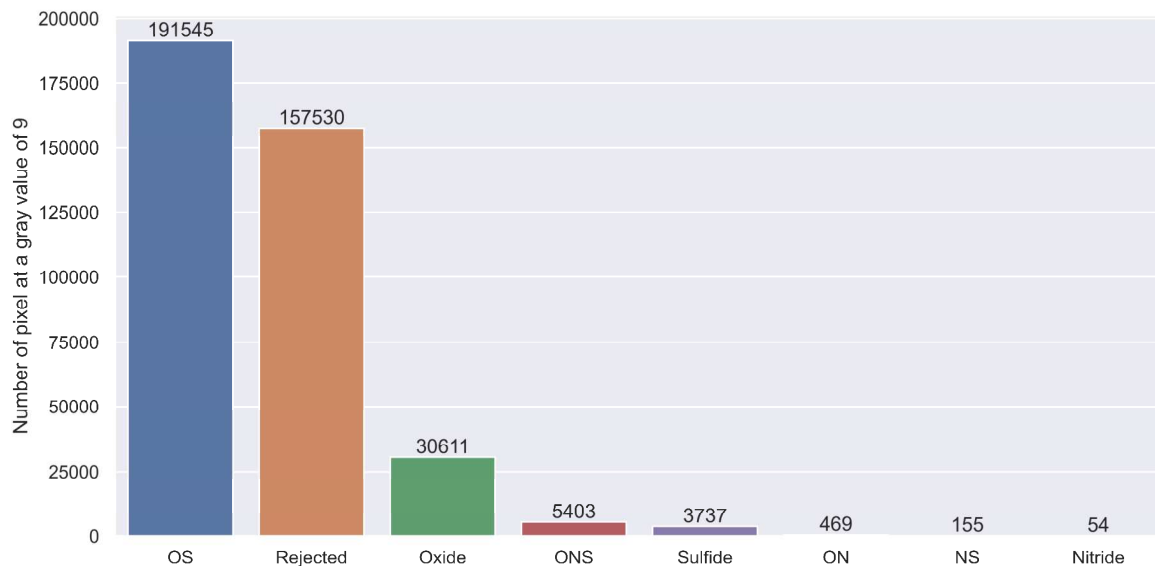


Figure 4-10: Number of pixels with a gray value of 9 from NMI classes in the dataset

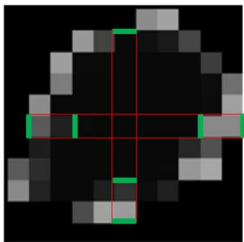
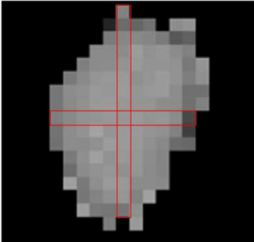
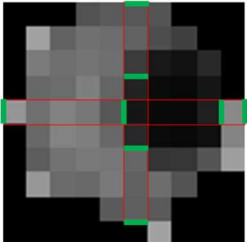
Another characteristic of different NMI classes in BSE images is the gray value gradient. A lot of image processing techniques exist for calculating gray value gradients and extracting associated features [37], but dealing with very small images is difficult due to the limited spatial

resolution and information content. Especially noise affects image quality, corrupts the information as well as distorts the image readability [38]. In this work, two approaches were used for representing the gray value gradient. The first one dealt with a manual calculation of how often gray values changed significantly in vertical and horizontal direction. Alternatively, a simpler approach using statistical parameters from gray value histograms as features was tested.

4.3.1.1 Vertical and Horizontal Gray Value Gradients

The different representation of oxides, sulfides, and OS in BSE images underlay the theory behind this approach. The gray value inside sulfides and oxides was expected to behave uniformly, while a more or less pronounced dark oxidic part influenced OS inclusions depending on the oxygen/sulfur ratio. Oxides showed a specialty with a gray ring surrounding the actual inclusion. The amount of significant gray value changes along vertical and horizontal direction as well as the amount of pixel with no significant change and their mean gray value provided a feature for machine learning. **Table 4-III** shows exemplary the calculation of these parameters on images of the three NMI classes. The row and column, which go through the center of the image, defined the horizontal and vertical direction as well as for calculation relevant pixels. Green lines represent exemplary a significant change of the gray value between two neighbored pixels. The number of these changes was expected to be the highest for OS and oxides and less common for sulfides. Distinguishing feature between OS and oxides were the mean gray values and number of pixels between two significant changes. Oxides only had the small gray ring, whereas OS consisted of significant more gray areas, especially with higher sulfur content.

Table 4-III: Comparing the results of the gray value gradients of oxide, sulfides, and OS

	Oxide	Sulfide	OS
Exemplary images			

A significant change of gray value occurred, if the difference between two neighbored pixels exceeded a threshold. For defining the positions of these changes in the observed row and column, binarization was used as an image processing technique. The application of this approach relied heavily on finding a suitable threshold for binarization. The threshold needed

to be in the gray value range between oxide and sulfide inclusions. Even though measurement parameters for all the observed steel grade remained the same across automated SEM/EDS analysis, the gray value distribution from sulfides differed significantly from each other, which is represented in **Figure 4-11**. Aside from the bearing steel, all other steel grades showed a similar gap between the peak at low and high gray values. It was necessary to ensure, that the binarization process did not split up sulfide inclusions. The gray value of oxides defined the lower limit of the threshold. After consideration of the different gray value distribution, the threshold was set to a value of 20.

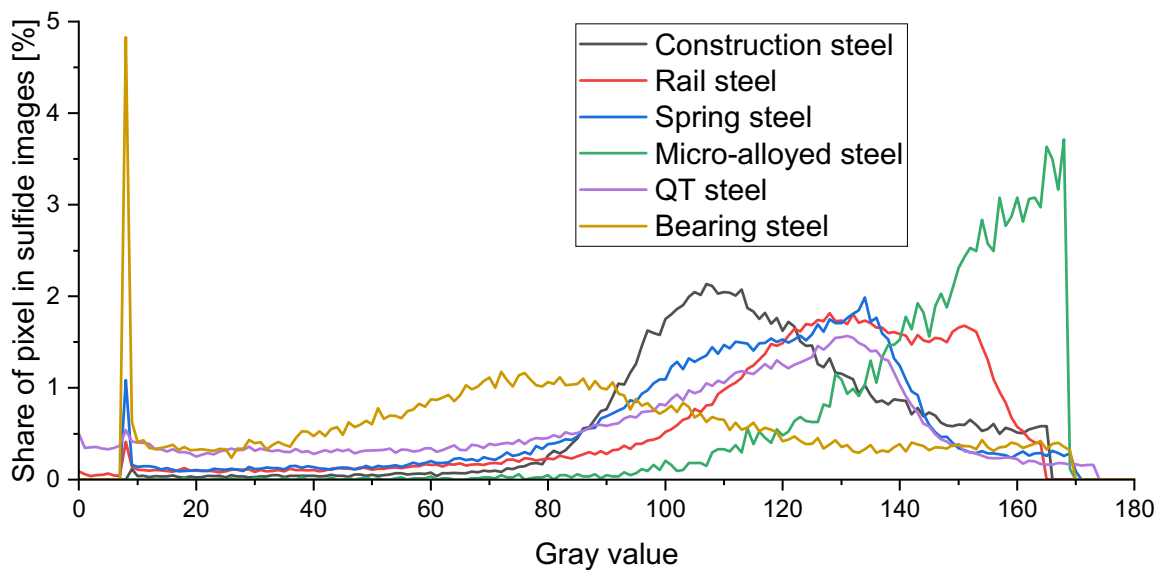


Figure 4-11: Summed up gray value distribution of every image in the respective steel grades (excluding the austenitic steel due to low number of sulfides)

In the bearing steel, 3,5 % of the pixels from sulfide BSE images had a gray value of 8. This indicates that vertical and horizontal gray value gradients was not applicable for this steel, because the described problem with splitting up sulfides occurred.

After binarization, the positions of the significant changes were calculated by comparing two neighbored pixel values. Finally, following parameters for both horizontal and vertical directions were defined:

- Number of pixels higher than the threshold divided by the length of the corresponding row or column
- Mean gray value of pixels higher than the threshold
- Number of pixels below the threshold divided by the length of the corresponding row or column
- Mean gray value of pixels below the threshold

If images only had one significant gray value change, as it is the case for the vertical direction in the exemplary oxide in **Table 4-III**, non-calculatable parameters were set to zero. Using this feature extraction method, 3600 input variables could be simplified into 8 features.

The quality of features, generated with this method, depended on the size of the images. The minimum inclusion length and breadth during automated SEM/EDS analysis was set to a value of 1 μm (3 pixels). Calculating the mentioned parameters inside a row or column with a length of 3 pixels led to results, which were not useable for machine learning. Limiting the input size of images evaluated by this method was expected to help generating high-quality features, but came at the cost of losing data. Especially in this dataset, most of the inclusions had small dimensions. The ECD distribution is showcased in **Figure 4-12** with a logarithmic scaled y-axis. 52 % of the NMIs had a ECD of 1 μm . 88 % were smaller than or equal to 3 μm (9 pixels).

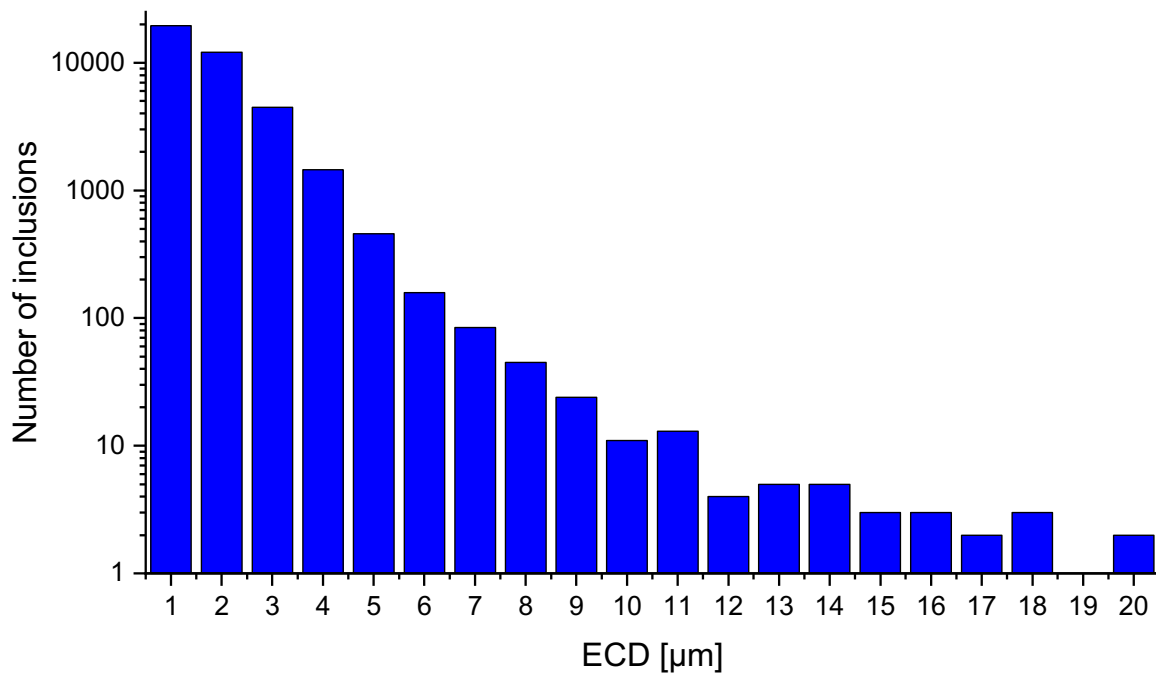


Figure 4-12: ECD distribution of the dataset

4.3.1.2 Statistical Parameter from Gray Value Histograms

An easier approach of defining features was using statistical parameters, such as mean, median, standard deviation, and quantiles from gray value histograms for describing the differences of the NMI classes. As stated by Piovesana et al. [39], for a stable calculation of mean and standard deviations in normative datasets, sample size of 50 is sufficient. Therefore, every image below a size of 50 pixels (approximately 2,5 μm ECD) could not be used for histogram evaluation because information was not sufficient enough and noise significantly

affected the mean. After sorting out inclusions lower than 50 pixels, 4965 OS, 1441 sulfides, 567 oxides, and 662 rejected data were evaluated. **Figure 4-13** shows the distribution of the mean image gray values respective their inclusion classes. The peak for oxide inclusions was at low mean gray values, whereas sulfide inclusions had a higher mean gray level. OS inclusions showed two peaks, one at lower MGV and one at higher MGV. Rejected data lay somewhere in between the oxide and sulfide peak. Even though the peaks could be distinguished from each other, a high image noise caused overlap between the NMI classes.

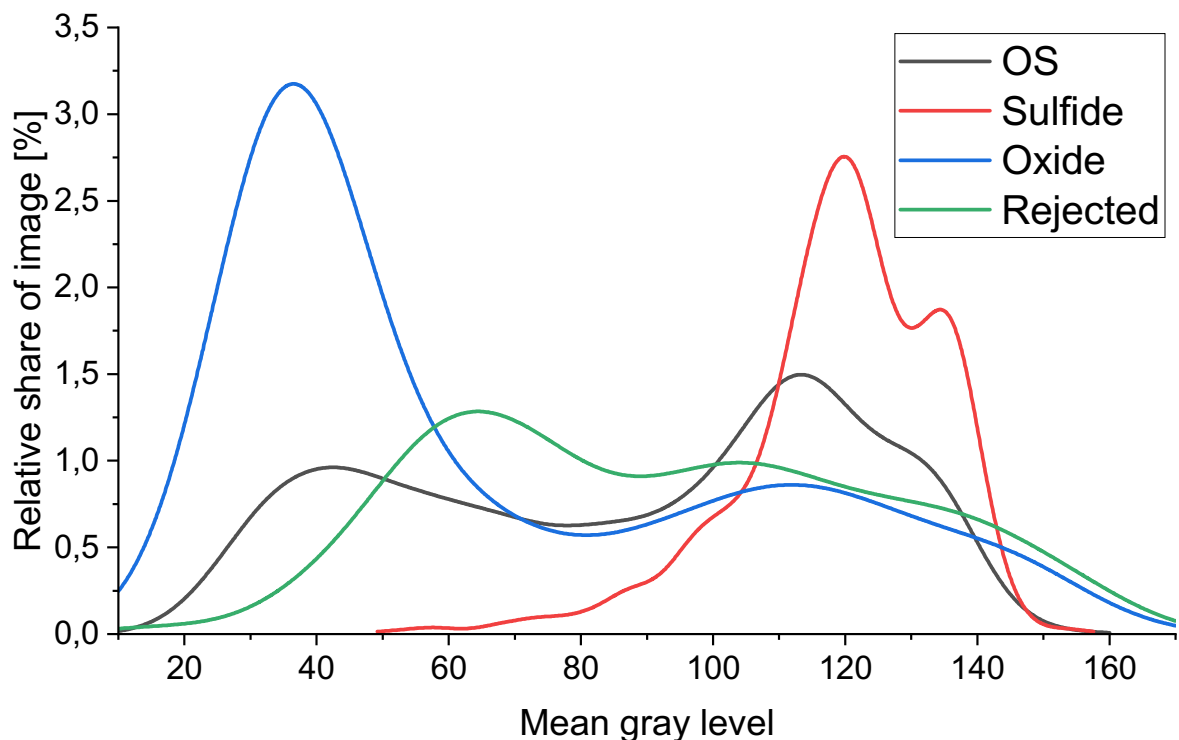


Figure 4-13: Relative share of images with a certain mean gray value

Figure 4-14 shows the calculated means, medians, as well as 25 % and 75 % quantile of the mean gray values from the images. Minimum and maximum values had no significant difference in the four presented classes. The overlap of the mean gray values in the 25 % to 75 % quantile led to the conclusion, that the amount of data points between the first and third quantile could not be used as a distinguishing factor and subsequently as feature for machine learning.

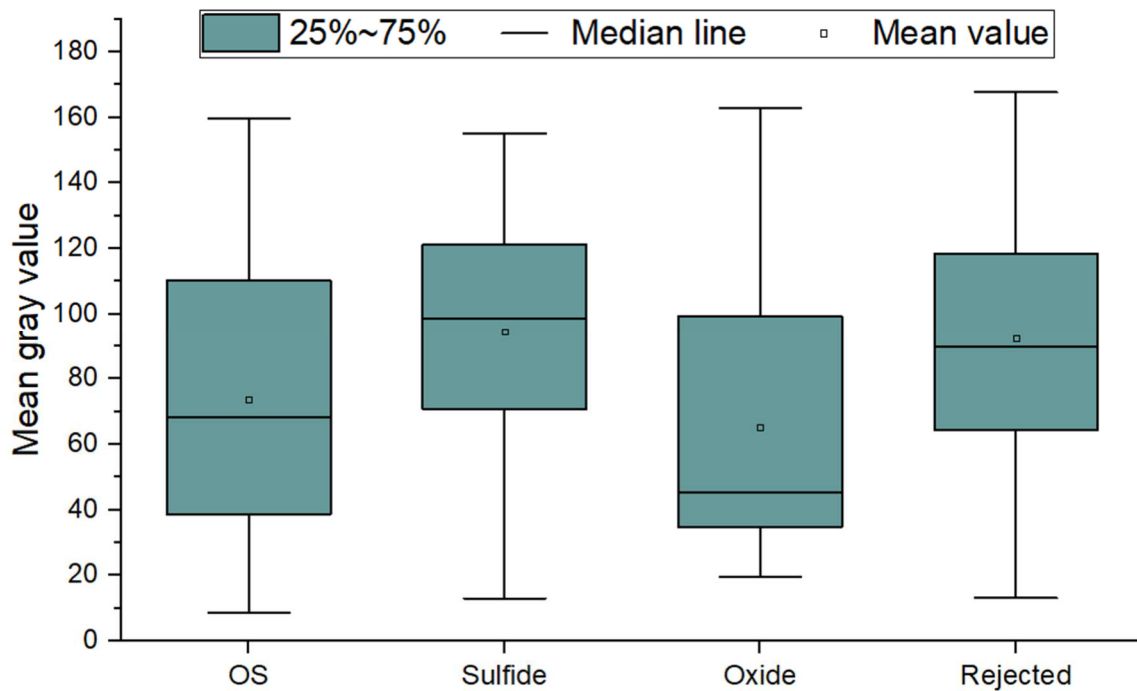


Figure 4-14: Median, mean, 25 % and 75 % quantiles of the mean gray values of all images

The median distribution of BSE images in the dataset is presented in **Figure 4-15**. Apart from sulfides, other inclusion classes show a peak at low median values. This peak is especially pronounced for the oxide class with 60 % relative share of images. The number of images at that median is comparably lower for OS and rejected class. In contrast, sulfides show a peak at higher median values, containing 20 % of sulfide images. The distribution between the two peaks is uniform with a low share of images at the respective median values and shows no difference between the NMI classes. Therefore, this statistical parameter is not suited as a feature for machine learning.

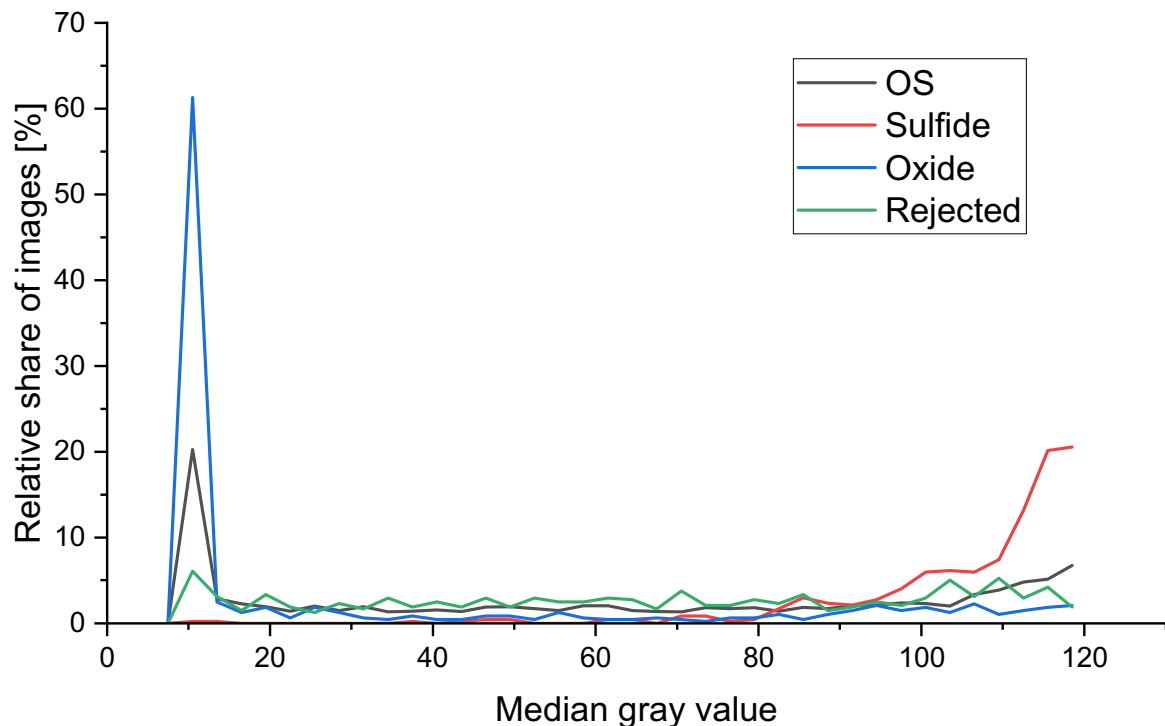


Figure 4-15: Relative share of images with a certain median gray value

4.3.2 Feature Selection of Geometric Parameters

Feature selection aims to identify a set of variables from the input data that effectively describes the underlying information while minimizing the impact of noise or irrelevant variables. Understanding the dependencies of these variables is important, due to the fact that highly correlated variables provide no further information about the classes and only increase the noise for the machine learning model. [33]

A correlation matrix was used to represent the relationship between the geometric features. The values inside this matrix can range from -1 to 1. Either a high or a low value leads to a linear correlation between two variables. If the value is 0, no correlation between two variables is present. **Figure 4-16** shows the correlation matrix of the geometric features from the dataset, which were calculated by Aztec software. Dark green fields represent a high correlation between the corresponding features, whereas for white fields no correlation could be found. The parameters 'Stage X', 'Stage Y' and 'Direction' had no measurable correlation with any other parameter. However, these three parameters needed to be dropped from the dataset because they contain no information about an inclusion, which can be used for classification. The parameter, which describe the size of an inclusion (area, ECD, breadth, length, perimeter)

were highly correlated with each other and can be combined to lower the number of features. The ECD was defined to represent the size of an NMI. Shape and aspect ratio of the inclusions showed some correlation within each other and with geometric dimensions such as length and perimeter. Due to the fact, that length and perimeter were already described by the inclusions' ECD, aspect ratio and shape represented features for machine learning. Furthermore, mean gray level was not correlated with any other parameter. In conclusion, the final dataset consists of the parameters ECD, aspect ratio, shape, and mean gray level.

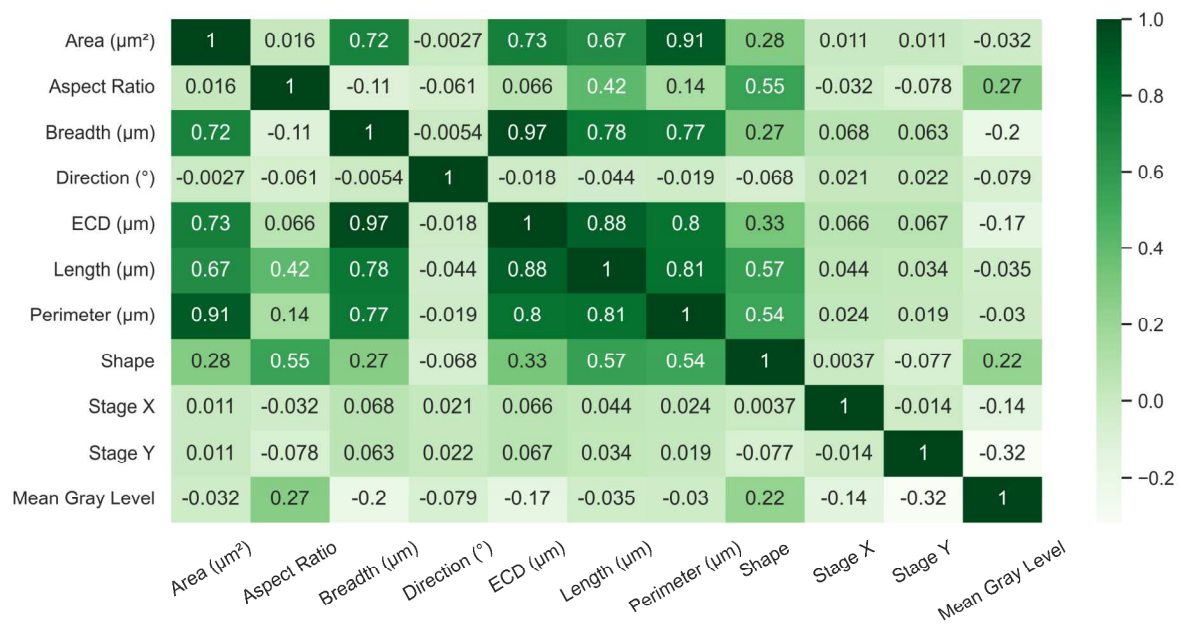


Figure 4-16: Correlation matrix of geometric parameters

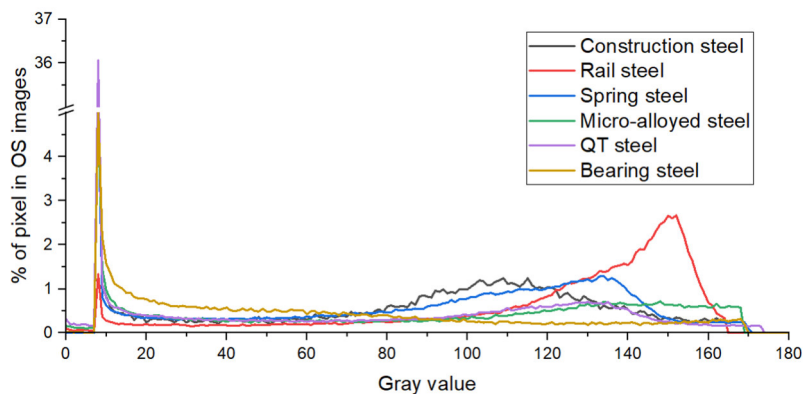
4.3.3 Comparing Different Steels

This chapter deals with the comparison of NMIs between the steel grades to evaluate if the dataset could be combined or needed to be split up. As already mentioned, a high inter-class variation led to problems during classification with machine learning models. **Table 4-IV** showcases the distribution of the different NMI classes (all together, oxide, sulfide, OS, rejected) in each steel grade with a short description of the characteristics. The diagrams were calculated directly with the gray value of pixels from all images.

Table 4-IV: Comparison of gray value distributions of the most important classes from the dataset

Pixel gray value distribution	Characteristics
All NMI classes	
	<p>Every steel showed a more or less pronounced peak at gray value 9 → influence of oxides</p> <p>Curves at higher gray value differed significantly from each other → influence of sulfides</p>
Oxides	
	<p>Very high peak at gray value 9 (up to 42 % pixel share in QT steel)</p> <p>Gray ring around oxides caused noise, which can be seen in alternating curve shapes at higher gray values.</p>
Sulfides	
	<p>Bearing steel was the only steel which showed a peak at gray value 9 for sulfides.</p> <p>Curves differed significantly from each other at higher gray values.</p>

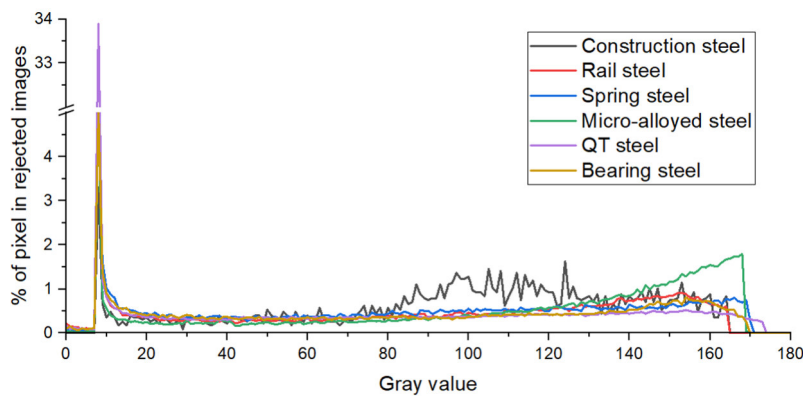
OS



Bearing, QT, and micro alloyed steel showed a peak at gray value 9.

Height of the peak at higher gray values depended on the steel.

Rejected



Behavior similar to oxides, but with a smaller peak at low gray value.

Considering the most important NMI classes in the dataset, distributions of gray values from sulfide images differed the most between the different steel grades. Bearing and micro alloyed steel showed significantly different curves compared to the other steels. Sulfides in the bearing steel peaked at low gray values, which was generally only the case for oxides. The second elevation started with a gray value of approximately 40 and resulted in a very widely ranged plateau, which ended at 130. Compared to the other steels, the gray level of sulfides was shifted to lower values. The peak of sulfides from the micro alloyed steel was located at very high gray values up to 170 and ended there abruptly. For evaluation of the different sulfide curves, the NMI type distribution may be considered. Classes are very broad defined and can lead to wrong conclusions about the characteristics and reasons behind the different shaped curves. **Figure 4-17** contains the sulfide-type distribution of the steels. Even though the gray value distribution differed significantly for the bearing and micro alloyed steel, type distribution was similar with mainly Mn-sulfides and Mn-Ca sulfides. This led to the conclusion, that the bearing and micro alloyed steel needed to be split from the dataset for machine learning.

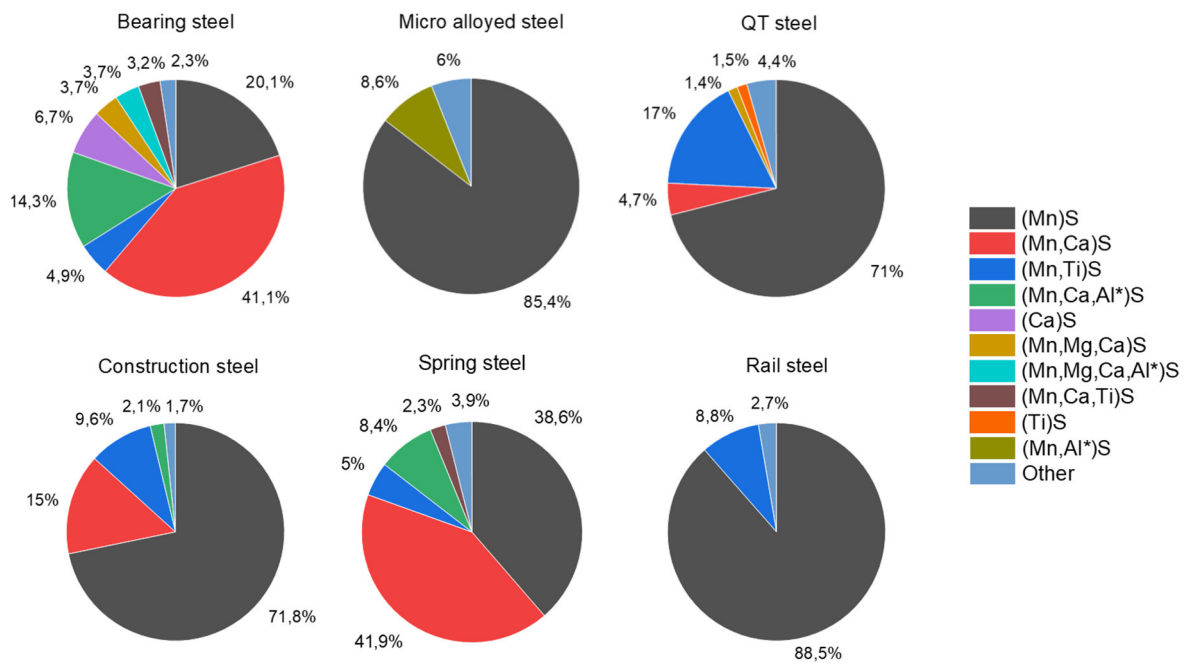


Figure 4-17: Sulfide-type distribution of the steels

5 Training and Evaluation of Machine Learning Models

To generate a basic understanding about the effectiveness of the proposed data preprocessing methods and the extracted features for this specific machine learning application, several datasets were created and used as training data for different machine learning models from the scikit-learn [40] python library.

5.1 Definition of the Training Sets

The discussed preprocessing methods led to overall eight created datasets from the construction-, rail-, spring-, QT- and austenitic steel:

- NMI-class database 1: Geometric features
- NMI-class database 2: Gradient features
- NMI-class database 3: Number of pixels with gray value of 9 (in graphics referred to as 'Nr. of pixels at gv 9')
- NMI-class database 4: Pixel features
- NMI-type database 1: Geometric features
- NMI-type database 2: Gradient features
- NMI-type database 3: Number of pixels with gray value of 9
- NMI-type database 4: Pixel features

The NMI-class databases consisted of four classes, namely OS, sulfide, oxide, and rejected data. As shown in **Figure 5-1**, these databases had a total number of 26690 inclusions. The previously mentioned features were extracted for each NMI.

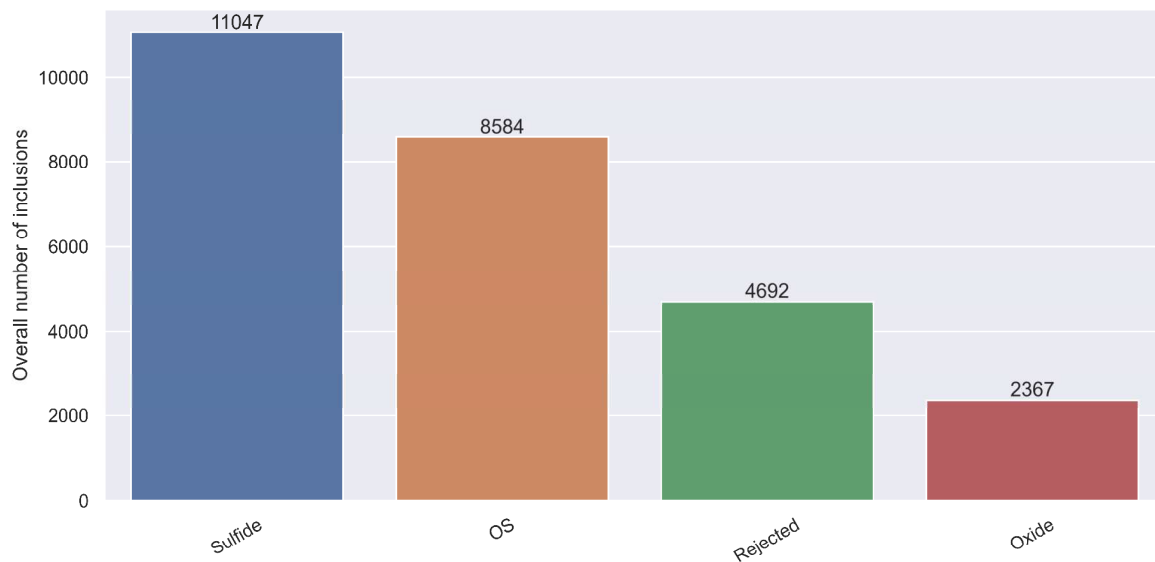


Figure 5-1: Class distribution of the NMI-class databases

Type specific labeling was used in the NMI-type databases, containing (Mn)S, not typified, SiC/matrix, (Mn,Ca)S, (Mn)O<S, (Al,Ca,Mn)OS, and (Mn,Ti)S. For these databases, the overall number of NMIs was 17388 (**Figure 5-2**).

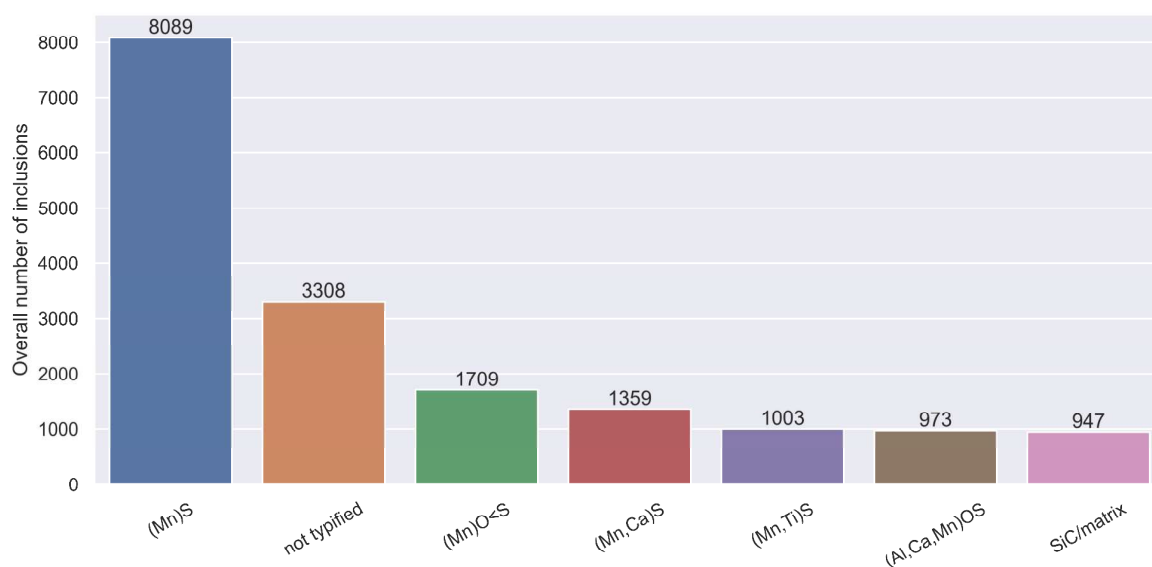


Figure 5-2: Type distribution of the NMI-type databases

5.2 Definition of the Machine Learning Models

The scikit-learn library provides various machine learning models. **Table 5-I** shows the different classifiers, which were used in this work for the classification task.

Table 5-I: Classifier models from scikit-learn library

Abbreviation	Classifier
SVM (Linear)	Support Vector Machine with Linear Kernel
SVM (Poly)	Support Vector Machine with Poly Kernel
SVM (RBF)	Support Vector Machine with RBF Kernel
SGD	Stochastic Gradient Decent
AB	AdaBoosting
BC	Bagging Classifier
GB	Gradient Boosting Classifier
HGB	Hist Gradient Boosting Classifier
RF	Random Forest

For comparison of the performance, different evaluation metrics (accuracy, precision, recall, ROC) were considered. Higher metric values represent a better model performance. K-fold cross validation with a k-value of 3 was used to train and evaluate the classifier. Following steps occur during this method:

- Data shuffling
- Data splitting in three folds (66 % training data, 33 % test data)
- Classifier training on each fold (test fold is left out)
- Calculation of average test accuracy and other metrics

The computational complexity of an SVM depends on the type of kernel function that is used. For linear kernel functions, the computational complexity is $O(n*m)$ where n is the number of training samples and m is the number of features. For non-linear kernel functions, such as the RBF or polynomial kernels, the computational complexity is $O(n^3)$ or higher, as it requires the calculation of the kernel matrix. A trial experiment for estimation of the training time was done by training a linear SVM with pixel features of the undersampled NMI-class database. This database had a dimensionality of 9468x3600. Training of the linear SVM took approximately 40 hours. Due to the computational complexity, training time would significantly rise in the case of oversampling, where the dimensionality of the database was 44188x3600. To minimize computational cost and training time, the Stochastic Gradient Decent (SGD) classifier substituted the SVM for the training with pixel features. The SGD classifier is a linear classifier, such as SVM, optimized by the SGD. This optimization process involves minimizing a loss function, which measures the error of the predictions compared to the true labels, and is particularly useful when dealing with large datasets, as it can effectively update the model parameters based on a subset of the data at each iteration. [41,42]

5.3 Comparison between Under- and Oversampling

The imbalance in the datasets had to be addressed due to the fact, that machine learning models lean more to the majority class and eliminate the minority class. Resampling techniques, such as over- and undersampling, are common approaches to make the dataset balanced. Undersampling involves reducing the number of instances or samples in the majority target class, while oversampling can be accomplished by generating new instances or duplicating existing ones in the minority class to increase their representation (**Figure 5-3**). [43]

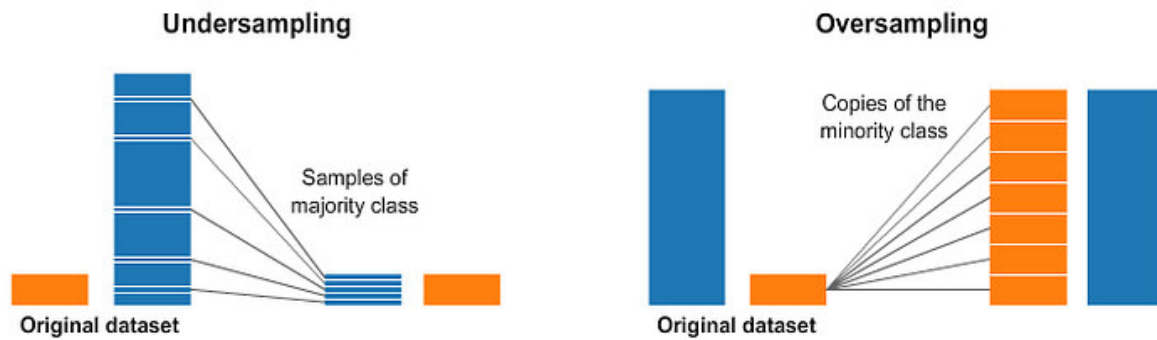


Figure 5-3: Under- and oversampling of a dataset [43]

Resampling techniques were utilized as the first experimental approach to address the present class imbalance in the dataset. The random oversampling algorithm, a non-heuristic approach, balances class distribution by generating new instances of the minority classes through random replication. The primary objective of the algorithm was to ensure that the model had sufficient data to learn the patterns and characteristics of the minority classes. It is important to note, that the oversampling techniques has its limitations. One such limitation is the potential risk of overfitting, which can arise due to the creation of identical replicas of minority class instances and causing the model to learn noise. This leads to a reduced generalization ability on unseen data. The undersampling technique aims to address class imbalance by reducing the number of instances in the majority class. In this experiment, undersampling was done by randomly selecting a subset of NMI from the dominant classes or types. By removing some of the instances from the majority classes, the distribution becomes balanced. However, this approach leads to a reduction in the amount of data available for training, which may impact model's ability to learn the full range of patterns and characteristics present in the data. [43]

5.3.1 Sampling Strategy 'not majority' and 'not minority'

In the first experiment, sampling strategies were set to 'not majority' and 'not minority', which resamples all classes except the majority or minority class, respectively. **Figure 5-4** summarizes the accuracy of the classifier trained on the over- and undersampled datasets with standard parameters. Z-scores were used to standardize the features in the datasets (Eq. 5-1).

$$Zscore = \frac{data\ point - mean}{standard\ deviation} \quad (Eq.5-1)$$

			Accuracy [%]								
			SVM (lin.)	SVM (poly)	SVM (RBF)	SGD	AB	BC	GB	HGB	RF
NMI-class database	Under-sampling	Geometric features	44,8	46,8	60,5	-	55,8	53,1	60,3	57,6	55,2
		Gradient	42,0	45,6	49,5	-	47,9	46,0	51,5	51,4	49,4
		Nr. of pixels at gv 9	30,6	25,4	31,8	-	37,1	37,4	37,4	37,1	37,2
		Pixel features	-	-	-	36,4	54,0	58,0	61,8	64,6	63,2
	Over-sampling	Geometric features	44,8	48,6	60,9	-	56,8	81,2	62,3	66,9	83,0
		Gradient	43,8	47,3	51,5	-	49,3	77,9	54,8	62,8	80,7
		Nr. of pixels at gv 9	31,8	25,9	33,5	-	38,1	38,6	38,6	38,4	38,6
		Pixel features	-	-	-	33,8	54,0	83,7	65,5	77,6	86,2
NMI-type database	Under-sampling	Geometric features	40,7	44,9	50,9	-	43,8	45,8	48,0	46,4	46,3
		Gradient	39,8	41,2	45,6	-	40,6	38,5	43,8	42,2	41,0
		Nr. of pixels at gv 9	17,5	18,3	25,5	-	25,6	25,7	25,7	25,7	25,7
		Pixel features	-	-	-	27,1	42,1	47,3	50,0	51,0	49,9
	Over-sampling	Geometric features	40,1	42,1	48,6	-	42,8	44,9	57,1	72,9	92,7
		Gradient	36,0	37,4	42,3	-	39,8	90,6	50,8	69,2	92,0
		Nr. of pixels at gv 9	15,8	16,6	24,0	-	25,5	25,5	25,6	25,5	25,5
		Pixel features	-	-	-	33,8	43,4	93,4	62,3	87,8	95,8

Figure 5-4: Accuracy of classifiers for under- and oversampling

The type of resampling technique did not influence the accuracy of the linear and kernel SVMs. Moreover, the accuracy of AdaBoosting and Gradient Boosting classifiers was not affected by using either under- or oversampling on the imbalanced datasets. The greatest increase in performance could be observed for the Bagging and Random Forest classifiers with a significantly higher accuracy for the oversampled NMI-class and -type databases. Only exception was the database containing the number of pixels with a gray value of 9 as a feature. To better understand the origin of the improved performance, other metrics such as precision and recall had to be considered. The precision is the fraction of correct predictions among the positive labels [42]. For example, a precision of 80 % for the class OS would mean, that 80 % of the predicted OS are actual OS and the other 20 % originate from other classes. Therefore, this metric gives information about how accurate the classifier is out of the predicted positive labels. The recall, or sensitivity, of a prediction vector is the fraction of true positive catches [42]. Continuing with the example from before, a recall of 80 % for the class OS means that 80 % of the actual OS from the whole database are predicted correctly. **Figure 5-5** shows these two calculated performance metrics of the Bagging classifiers trained on the NMI-class databases. The color distribution refers to the minimum and maximum value in the precision or recall column.

		Precision [%]				Recall [%]			
		OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Under-sampling	Geometric features	52,2	50,7	48,1	62,3	57,5	48,3	44,7	63,2
	Gradient	43,9	44,7	42,4	54,4	46,3	45,1	37,5	57,2
	Nr. of pixels at gv 9	42,6	51,6	41,3	34,1	48,2	10,2	0,4	90,7
	Pixel features	54,9	55,6	55,4	68,9	61,2	53,6	50,2	69,2
Over-sampling	Geometric features	75,9	86,7	81,6	79,9	75,1	96,7	84,9	68,4
	Gradient	70,3	85,5	79,1	74,6	67,7	96,9	83,5	63,2
	Nr. of pixels at gv 9	44,1	60,4	68,8	34,4	49,7	12,4	0,6	91,2
	Pixel features	76,5	90,0	85,1	82,6	76,2	97,0	87,4	74,2

Figure 5-5: Precision and recall of Bagging classifiers trained on over- and undersampled NMI-class databases

Except for the NMI-class database number of pixels with a gray value of 9, oversampling increased the precision and recall of every class. Even though the number of sulfides did not change during resampling, the classifier could achieve a better performance within this class. One possible explanation for this observation is, that the undersampled databases may not have contained sufficient information for effective pattern detection. By learning more about the other classes in the dataset, the algorithm improved its ability to accurately classify sulfides, resulting in a better performance. The same characteristic showed the Random Forest classifier, illustrated in **Figure 5-6**.

		Precision [%]				Recall [%]			
		OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Under-sampling	Geometric features	54,4	52,9	50,0	62,0	58,6	48,8	44,8	68,4
	Gradient	48,9	47,2	44,7	54,7	46,9	46,1	41,1	63,2
	Nr. of pixels at gv 9	42,5	50,7	42,9	34,1	47,7	10,5	0,5	90,7
	Pixel features	60,5	61,6	58,7	70,0	61,8	58,0	54,2	77,8
Over-sampling	Geometric features	78,4	88,9	84,1	79,9	76,0	97,6	86,1	72,6
	Gradient	75,3	88,9	81,9	76,2	69,6	97,4	86,3	70,5
	Nr. of pixels at gv 9	44,2	60,5	69,9	34,4	49,5	12,7	0,7	91,5
	Pixel features	80,7	94,5	87,2	82,1	77,5	97,1	88,9	81,5

Figure 5-6: Precision and recall of Random Forest classifiers trained on over- and undersampled NMI-class databases

5.3.2 Alternative Sampling Strategy

Due to the high number difference between the majority and minority classes in the databases, an alternative sampling strategy was tested. Overfitting is a problem with oversampled datasets, especially if the minority class has only a small share. To achieve the same number of instances as sulfides, oxides had to be replicated five times for oversampling. This is even higher for SiC/matrix particles, as they had to be copied eight times to generate a balanced dataset. To address the issue of excessively high replica counts, an alternative sampling strategy was used that combines both over- and undersampling techniques. This approach aimed to balance the class distribution by generating new instances of the minority class through oversampling while simultaneously reducing the number of instances in the majority class through undersampling. For the NMI-class databases, the number of rejected particles was the reference point. Applying the alternative sampling strategy resulted in an equal distribution, with each class containing 4692 inclusions. The same process was implemented for the NMI-type databases, where the number of '(Mn)O<S' particles defined the position of equality. **Figure 5-7** shows exemplary the functionality of the alternative sampling strategy on the NMI-type distribution.

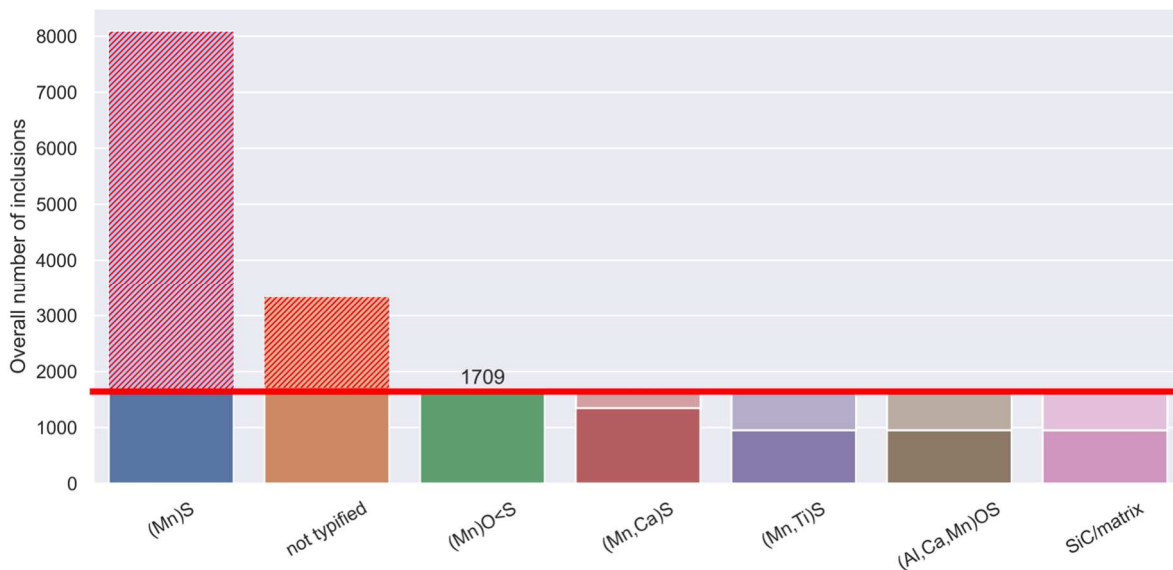


Figure 5-7: Result of the alternative sampling strategy on the NMI-type distribution

As shown in **Figure 5-8**, compared to oversampling the alternative sampling strategy resulted in a lower precision and recall for the Bagging classifier trained on NMI-class databases. Artificially increasing the number of oxides and OS by a huge amount, which was the case in oversampling, may have led to falsified performance metrics due to overfitting the training data.

Alternative sampling performed better than undersampling and represents a compromise between the number of replicas in the dataset, which cause a higher chance of overfitting, and classifier performance. For following experiments in the next chapters, the alternative sampling approach was implemented to balance the databases.

		Precision [%]				Recall [%]			
		OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Under-sampling	Geometric features	52,2	50,7	48,1	62,3	57,5	48,3	44,7	63,2
	Gradient	43,9	44,7	42,4	54,4	46,3	45,1	37,5	57,2
	Nr. of pixels at gv 9	42,6	51,6	41,3	34,1	48,2	10,2	0,4	90,7
	Pixel features	54,9	55,6	55,4	68,9	61,2	53,6	50,2	69,2
Alternative sampling	Geometric features	59,8	67,5	57,7	65,4	61,6	78,3	47,8	64,1
	Gradient	51,4	65,1	49,6	57,9	50,2	76,5	43,1	56,6
	Nr. of pixels at gv 9	43,4	58,1	72,8	34,5	49,0	11,9	0,5	91,6
	Pixel features	61,8	71,6	65,0	72,1	63,8	81,8	55,8	69,8
Over-sampling	Geometric features	75,9	86,7	81,6	79,9	75,1	96,7	84,9	68,4
	Gradient	70,3	85,5	79,1	74,6	67,7	96,9	83,5	63,2
	Nr. of pixels at gv 9	44,1	60,4	68,8	34,4	49,7	12,4	0,6	91,2
	Pixel features	76,5	90,0	85,1	82,6	76,2	97,0	87,4	74,2

Figure 5-8: Comparison between alternative-, under-, and oversampling in the NMI-class databases with the performance of Bagging classifier

5.4 Comparison between Class- and Type Labels

In this experiment, the comparison of the classifier performance between the different databases was carried out to evaluate the influence of intra-class variance. The achieved accuracy of the classifier is shown in **Figure 5-9**. When training with NMI-class databases, the classification task involved four classes. In contrast, training with NMI-type databases requires a six-class classification. The increased accuracy of classifiers trained on NMI-class databases, by approximately 7-10 %, was a result of the task being less challenging due to the smaller number of classes to classify. When considering, that the easier task is the four-class classification, both types of databases performed similarly. These findings suggest that there was no significant intra-class variance present in the dataset, allowing classifiers to perform on the same relative level across different labelling methods. Due to the higher accuracy scores, the NMI-class databases were evaluated more detailed in the following chapters.

		Accuracy [%]								
		SVM (lin.)	SVM (poly)	SVM (RBF)	SGD	AB	BC	GB	HGB	RF
NMI-class database	Geometric features	43,5	48,2	60,7	-	56,5	62,8	61,5	62,0	64,9
	Gradient	40,6	47,4	50,5	-	48,4	56,8	53,6	56,6	60,0
	Nr. of pixels at gv 9	30,9	25,7	33,0	-	38,3	38,4	38,4	38,3	38,4
	Pixel features	-	-	-	36,1	53,8	67,0	63,7	70,1	71,2
NMI-type database	Geometric features	36,7	38,6	45,6	-	41,5	55,7	49,2	53,5	57,0
	Gradient	36,1	36,8	40,7	-	39,3	51,8	44,6	50,0	54,5
	Nr. of pixels at gv 9	18,0	16,9	19,5	-	20,8	20,9	20,9	20,9	20,9
	Pixel features	-	-	-	27,0	40,0	58,8	52,8	63,3	63,2

Figure 5-9: Comparison between class- and type-labelling

5.5 Bagging Classifier and Random Forest Classifier

As can be seen from the previous figures, Bagging classifier and Random Forest classifier showed the best performance in the datasets. This chapter provides a detailed overview of the functionality, achievable results, and further evaluation metrics for these two classifiers. Furthermore, different fine-tuning methods will be compared and the importance of features discussed.

Bagging Classifier and Random Forest Classifier are ensemble methods, which aggregate the predictions of a group of predictors to get more accurate results than the best individual predictor. This technique is called ensemble learning. The bagging approach includes the usage of the same algorithm for every predictor and the training on different random subsets of the training set. Sampling is performed with replacement, referred to as bootstrap aggregating, which allows training instances to be sampled several times for the same predictor. **Figure 5-10** represents the sampling and training process. After training all predictors, the ensemble makes a prediction for a new instance by simply aggregating the predictions of all predictors. For a classification task, the aggregation function is typically the statistical mode (most frequent prediction). [44]

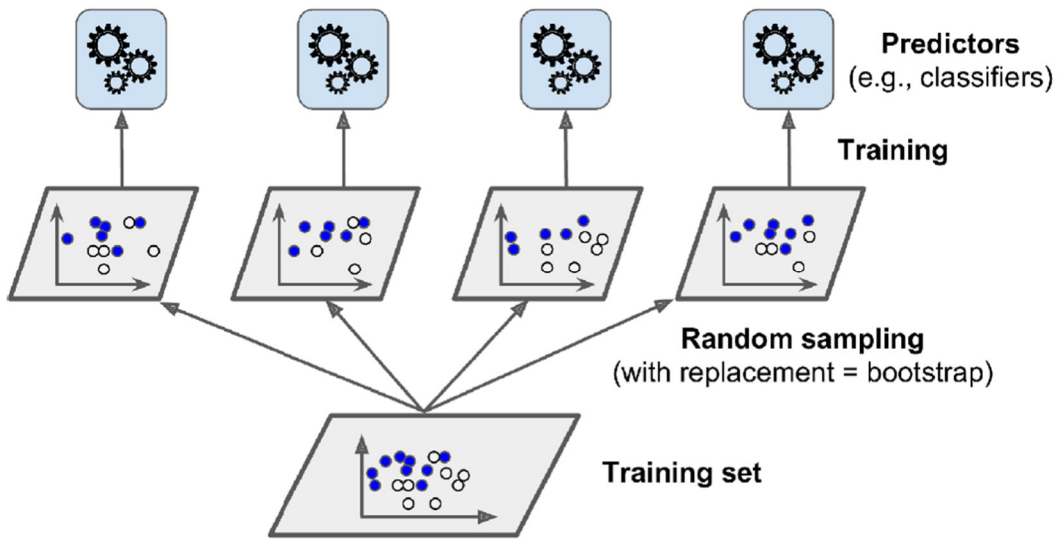


Figure 5-10: Sampling and training process of Bagging classifiers [44]

Random Forest classifier is an ensemble of Decision Trees, generally trained with the bagging method. The max size of samples is typically set to the size of the training set. In machine learning, a Decision Tree is a type of predictive model that makes predictions by traveling along a tree structure from the root node to a leaf. At each node on the root-to-leaf path, the model selects a successor child based on a split of the input space. The splitting process is based on one of the features or on a predefined set of splitting rules. A leaf contains a specific label. **Figure 5-11** shows an example of a decision tree. [42]

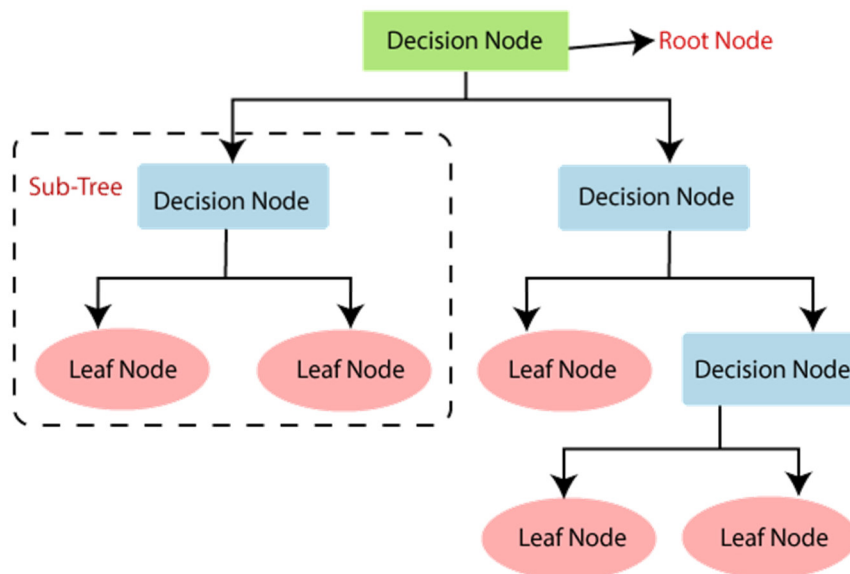


Figure 5-11: Schematic representation of a Decision Tree [45]

5.5.1 Comparison of the Performance between different NMI-Features

Figure 5-12 shows the precision and recall values for the Bagging classifier and the Random Forest classifier using different types of features. Both models were trained with standard parameters from the scikit-learn library. In general, the RF classifier outperformed the BC in terms of both precision and recall. Specifically, the RF classifier achieved higher precision and recall scores than the BC classifier for all feature types, except for the number of pixels with a gray value of 9, where the RF classifier had similar scores. Furthermore, the achieved results led to the conclusion that some feature types were more informative than others for classification. For example, the pixel features had the highest precision and recall scores for the BC and RF classifiers. In contrast, the feature number of pixels with a gray value of 9 had the lowest precision and recall scores for both classifiers and using it independently is not recommended. For a further experiment, the geometric features and the number of pixels with a gray value of 9 were combined into one dataset. However, this led to no performance increase. Geometric, gradient, and pixel features were used to enhance the performance of Bagging classifier and Random Forest classifier with fine-tuning of hyperparameters.

	Bagging classifier								Random Forest classifier							
	Precision				Recall				Precision				Recall			
	OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Geometric features	59,9	67,6	58,1	65,8	61,8	77,5	48,5	63,4	61,8	71,1	60	65,5	63,5	79,1	48,3	68,3
Gradient features	51,1	65,4	48,7	57,7	49,8	76,9	43,1	57	56,7	69,2	53,8	58,6	50,7	78	47,3	63,8
Nr. of pixels at gv 9	43,4	58,7	72,2	34,5	49,9	11,4	0,5	91,6	43,5	58,9	71,8	34,5	49,3	11,7	0,5	91,6
Pixel features	61,4	71	63,6	71,3	63	80,8	55,1	69,4	65,7	79,3	68	71,4	65,1	81,1	61,9	77,6

Figure 5-12: Precision and recall for BC and RF trained on alternative sampled NMI-class databases

5.5.2 Fine-Tuning

Fine-tuning of model parameters is the process of adjusting the hyperparameters of machine learning models in order to optimize its performance. Hyperparameter are set by the operator before training the model and not learned from the data. Examples of hyperparameters include the learning rate, batch size, as well as regularization and depend on the classifier. The scikit-learn library provides different strategies for fine-tuning. 'GridSearchCV' includes manual definition of the values for hyperparameters, which are evaluated. All the possible combinations are tested, and the best result will be plotted. When the number of combinations to explore is limited, the grid search method proves to be effective. However, in cases where the search space for hyperparameters is extensive, 'RandomizedSearchCV' is the preferable method. This technique assesses a fixed number of random combinations by randomly choosing a value for each hyperparameter. Using random search has two benefits. It explores

for every iteration different values for each hyperparameter instead of just a few combinations, as it is the case with the grid search approach. Furthermore, computing budget can be controlled easier by setting the number of iterations. [44]

5.5.2.1 Fine-Tuning of the Bagging classifier

In scikit-learn, Bagging Classifier samples by default training instances with replacement (`bootstrap=True`). On average, 63 % of the training instances are sampled for each predictor, whereas the remaining 37 %, which are called out-of-bag (oob) instances, are not used for sampling. As the predictor never comes across the oob instances during training, evaluation of the model can be done without requiring a separate validation set. Furthermore, the number of predictors (default value of 10) can be adjusted as well as the maximum number of samples, which control the size of the subsets. Sampling of features can also be implemented in the Bagging classifier by controlling the two hyperparameters '`max_features`' and '`bootstrap_features`'. Using sampling for training instances and features is called Random Patches method and particularly useful for training with high-dimensional data such as images. [44,46,47]

Grid search was utilized to determine the optimum number of estimators for Bagging classifiers. The size of the dataset represented the number of drawn samples. **Figure 5-13** illustrates the correlation between the accuracy and the number of estimators for Bagging classifiers trained on geometric and gradient features. The accuracy of both features started to rise at the beginning and then flattened out at approximately 100 used estimators. In case of geometric features the accuracy approached 66 %, whereas for gradient features the value was at a lower level of 60 %.

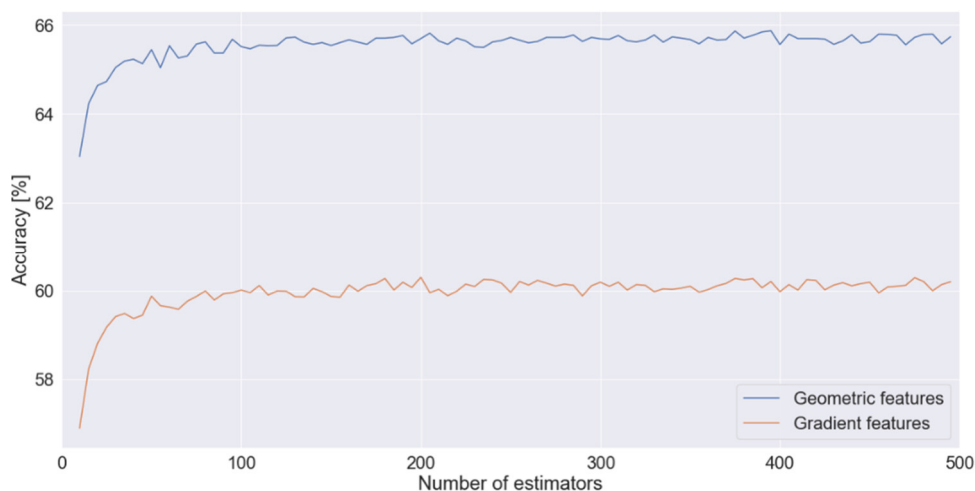


Figure 5-13: Correlation between accuracy and number of estimators for the Bagging classifier

For pixel features, the Random Patches method was tested, using randomized search by defining upper and lower limits for the number of estimators and maximum features. It was necessary to set the hyperparameter 'bootstrap_features' to a value of true. **Figure 5-14** shows a scatter plot, summarizing the results of the randomized search by illustrating the achieved accuracy with a given number of used estimators and maximum features. Increasing the number of used features and estimators up to a certain point led to a better performance. The plateau of the accuracy was at 72 %. This accuracy level was achievable by a relative low number of maximum features and high number of estimators and vice versa. For example, using 100 estimators only needed 300 features to get to the accuracy plateau. The information from the padded BSE images was compressible from 3600 input variables to a significantly smaller number, depending on the number of estimators from the Bagging classifier. Theoretically, reaching the highest accuracy could also have been achieved by using all features and a low number of estimators. However, this approach would lead to a significant higher time needed for training and the possibility to learn unimportant image noise. Hyperparameters for the final comparison between before and after fine-tuning were set for training on geometric and gradient features to:

- Number_estimators = 100

and for training on pixel features:

- Number_estimators = 100
- Max_features = 400
- Bootstrap_features = True

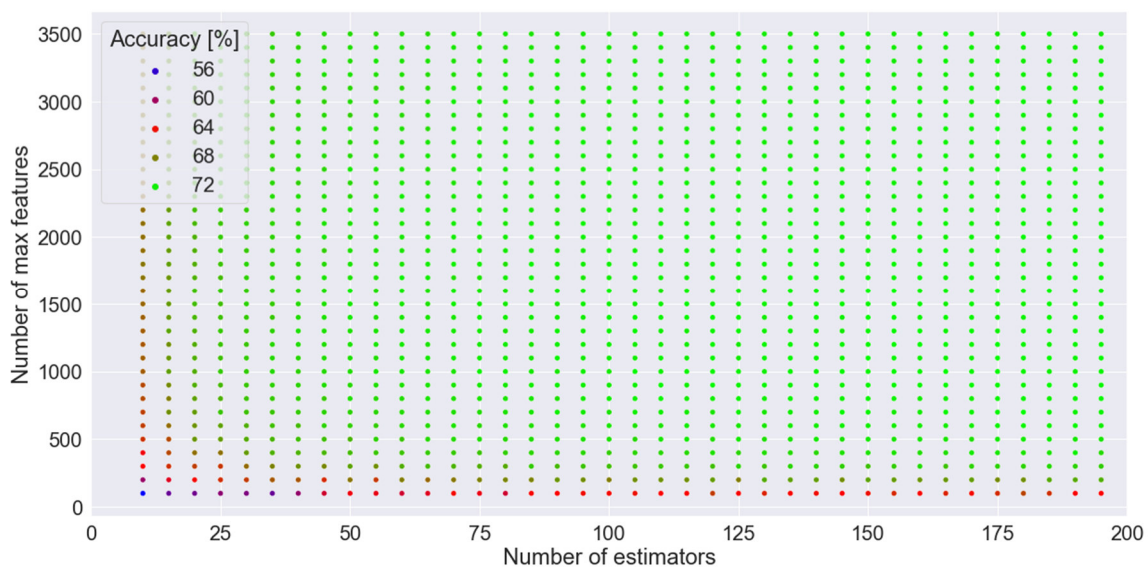


Figure 5-14: Result of using the Random Patches Method on pixel features with randomized search

A final comparison between the results of the Bagging classifier before and after fine-tuning is given in **Figure 5-15**. The table shows the precision and recall values of the four classes oxide, OS, rejected, and sulfide. For each class, there was an improvement in precision and recall after fine-tuning the model. The highest precision increase, up by 12 %, could be observed for oxides during classification with pixel features. Important regarding this high precision increase is, that the corresponding recall value did not change and therefore an overall better classifier performance was achievable. Furthermore, the recall for the rejected class rose by 14,2 % and the precision by 4,3 %. For geometric and gradient features the performance increase was not as significant as for the pixel features, but overall, the accuracy was still about 2 % higher than before fine-tuning. Especially the sulfide class benefited in these two types of features from the fine-tuning process by showing a higher recall.

		Accuracy [%]	Precision [%]				Recall [%]			
			OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Geometric features	Before fine-tuning	62,8	59,9	67,6	58,1	65,8	61,8	77,5	48,5	63,4
	After fine-tuning	65,1	62,7	70,6	60,0	65,7	62,1	79,0	50,7	68,6
	Difference	2,3	2,8	3,0	1,9	-0,1	0,3	1,5	2,2	5,2
Gradient features	Before fine-tuning	56,8	51,1	65,4	48,7	57,7	49,8	76,9	43,1	57,0
	After fine-tuning	59,1	55,3	68,1	52,7	58,1	49,1	77,4	47,0	62,9
	Difference	2,3	4,2	2,7	4,0	0,4	-0,7	0,5	3,9	5,9
Pixel features	Before fine-tuning	67,0	61,4	71,0	63,6	71,3	63,0	80,8	55,1	69,4
	After fine-tuning	72,0	68,0	83,6	67,9	71,4	64,4	80,9	69,3	77,4
	Difference	5,0	6,6	12,6	4,3	0,1	1,4	0,1	14,2	8,0

Figure 5-15: Achieved performance increase with fine-tuning the Bagging classifier’s hyperparameter

5.5.2.2 Fine-Tuning of the Random Forest classifier

In scikit-learn, the Random Forest classifier combines nearly all of the hyperparameter of the Decision Tree classifier, which control how trees are grown, and Bagging classifier. Following hyperparameter were adjusted during fine-tuning with random search:

- Number of estimators (n_estimators)
- Minimum samples split, which defines the minimum number of samples required to split an internal node (min_samples_split)
- Minimum samples leaf, which defines the minimum number of samples required to be at a leaf node (min_samples_leaf)
- Maximum features to consider when looking for the best split (max_features)
- Maximum number of levels in each decision tree (max_depth)
- Sampling techniques:

- With replacement (Bagging) → bootstrap = True
- Without replacement (Pasting) → bootstrap = False

Table 5-II summarizes the settings of the hyperparameter, which resulted in the best performance for a randomized search with 1000 iterations.

Table 5-II: Hyperparameter settings for Random Forest classifier

Hyperparameter	Geometric Features	Gradient Features	Pixel Features
n_estimators	200	200	200
min_samples_split	5	2	5
min_samples_leaf	1	1	1
max_features	'sqrt'	'sqrt'	500
max_depth	None	20	80
bootstrap	True	False	False

Figure 5-16 contains the results of the Random Forest classifier after fine-tuning and shows the performance increase compared to standard parameters. For geometric and gradient features, the accuracy was slightly better after fine-tuning, mainly due to higher precision scores for the OS and oxide class as well as higher recall score for sulfides. Regarding pixel features, the accuracy could be increased by 2 %. Especially the recall score for the rejected class improved during fine-tuning.

		Accuracy [%]	Precision [%]				Recall [%]			
			OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Geometric features	Before fine-tuning	64,9	59,9	67,6	58,1	65,8	61,8	77,5	48,5	63,4
	After fine-tuning	65,6	64,4	71,1	60,3	65,7	63,3	76,1	52,6	70,6
	Difference	0,7	4,5	3,5	2,2	-0,1	1,5	-1,4	4,1	7,2
Gradient features	Before fine-tuning	60,0	56,7	69,2	53,8	58,6	50,7	78,0	47,3	63,8
	After fine-tuning	60,4	57,0	72,3	54,7	56,9	50,4	73,9	48,4	68,6
	Difference	0,4	0,3	3,1	0,9	-1,7	-0,3	-4,1	1,1	4,8
Pixel features	Before fine-tuning	71,2	65,7	79,3	68,0	71,4	65,1	81,1	61,9	77,6
	After fine-tuning	73,1	68,1	81,9	68,1	73,9	66,1	80,3	68,3	77,5
	Difference	1,9	2,4	2,6	0,1	2,5	1,0	-0,8	6,4	-0,1

Figure 5-16: Achieved performance increase with fine-tuning the RF classifier's hyperparameter

Concluding the fine-tuning, Bagging classifier and Random Forest classifier showed improved performances. The optimum hyperparameter settings had more impact on the Bagging classifier. Both classifier performed the classification based on pixel features after fine-tuning on a similar level at 72 % to 73 % accuracy.

5.5.3 Feature Importance

Random Forest classifier are able to measure the relative importance of each feature by looking at how much the tree nodes that use the corresponding features reduce impurity on average [44]. After training with fine-tuned hyperparameters, scikit-learn computed the feature importance scores automatically. For geometric features, following scores were calculated:

- Aspect Ratio: 0,163
- ECD: 0,213
- Shape: 0,178
- Mean Gray Level: 0,446

The biggest impact on decision-making of the Random Forest classifier had the mean gray level of NMIs. The ECD represented the second most important feature. Aspect ratio and shape influenced the decision on a similar low level.

In case of gradient features, following feature importance results were evaluated:

- Row features:
 - Number of pixels higher than the threshold: 0,121
 - Mean gray value of pixels higher than the threshold: 0,258
 - Number of pixels below the threshold: 0,057
 - Mean gray value of pixels below the threshold: 0,062
- Column features:
 - Number of pixels higher than the threshold: 0,121
 - Mean gray value of pixels higher than the threshold: 0,259
 - Number of pixels below the threshold: 0,058
 - Mean gray value of pixels below the threshold: 0,062

Comparing the origin of features' extraction (row or column), the influence was very similar. As already observed with geometric features, the mean gray level showed the highest impact on decision-making. In case of gradient features, only the mean gray value of pixels higher than the threshold had a high influence, whereas the importance of the mean gray value of pixels lower than the threshold was almost neglectable.

Regarding pixel features, Random Forest classifier learned to focus on the middle part of BSE images (**Figure 5-17**). In the NMI-class dataset, most of the inclusions were in the range of 1 μm to 2 μm ECD, which is somewhere between 9 and 18 pixels. These 9 to 18 pixels are located in the center of **Figure 5-17**. Therefore, introducing the padding process, which added

the black pixels around the actual BSE images to ensure the size of 60x60 px, did not act as influence or distracting noise for the Random Forest classifier.

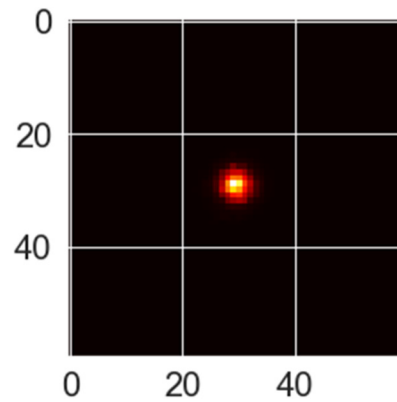


Figure 5-17: Feature importance of pixel from BSE images

5.5.4 ROC Curves

The receiver operating characteristics (ROC) curve plots the true positive rate, other name for recall, against the false positive rate, which is the ratio of negative instances that are incorrectly classified as positive [44]. The top left corner of ROC curves represents the ideal point, where the false positive rate is zero and the true positive rate one. However, reaching this point is not realistic as it represents a perfect classifier. Further important characteristics of ROC plots are the area under the curve and the steepness, since it is ideal to maximize the true positive rate while minimizing the false positive rate. **Figure 5-18** gives a schematic representation of a ROC curve. [48]

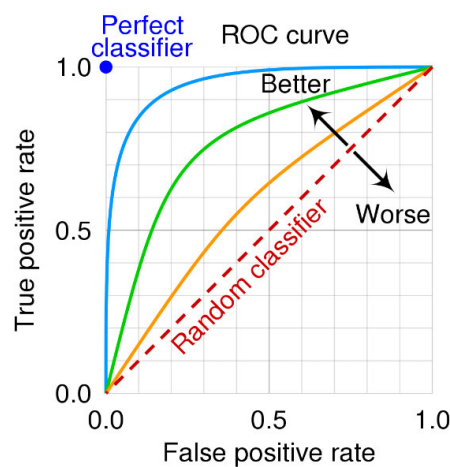


Figure 5-18: Exemplary ROC curve [49]

ROC curves are typically used for binary classification. In the case of multiclass classification, true and false positive rate are obtained after binarizing the output. The one-vs-rest scheme compares each class against all the others and was used for evaluating the ROC curves of sulfide, oxide, rejected, and OS. **Figure 5-19** describes the ROC curves of the Bagging classifier trained on geometric features. The classifier detected oxides better than other classes due to the steep slope of the 'Oxide vs Rest' curve. True positive rate increased for oxides very fast, while the false positive rate stayed at a very low value. In comparison to oxides, the classifier failed to achieve a similar false positive rate for other classes, if a particular true positive rate was targeted.

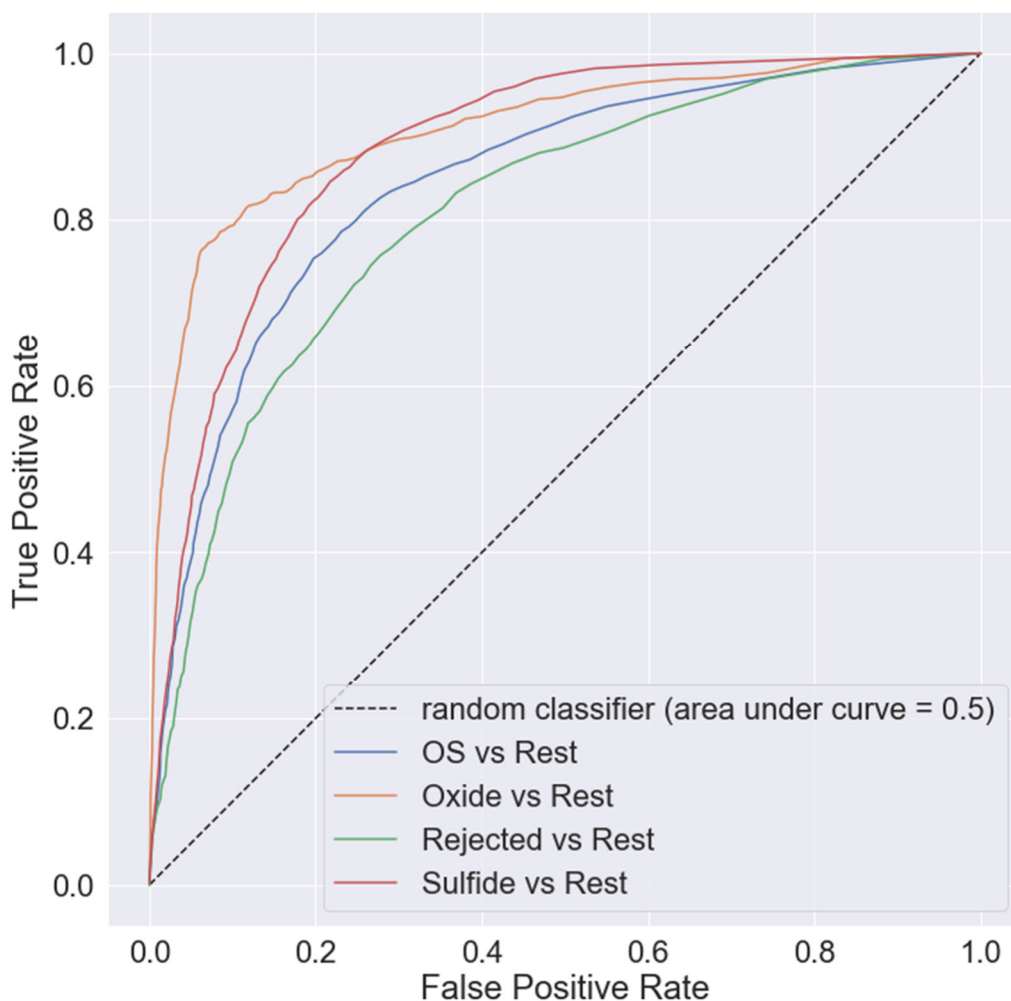


Figure 5-19: ROC curves of Bagging classifier trained on geometric features

Calculating the ROC curves for training of the Bagging classifier on pixel features leads to **Figure 5-20**. Compared to geometric features, training on pixel features resulted in a higher area under the curves for every class. This is explainable by the overall better achieved accuracy for the classification directly with BSE images. The ROC curve of 'Rejected vs Rest'

matches the shape of the 'OS vs Rest' curve. This was not the case for the Bagging classifier trained on geometric features.

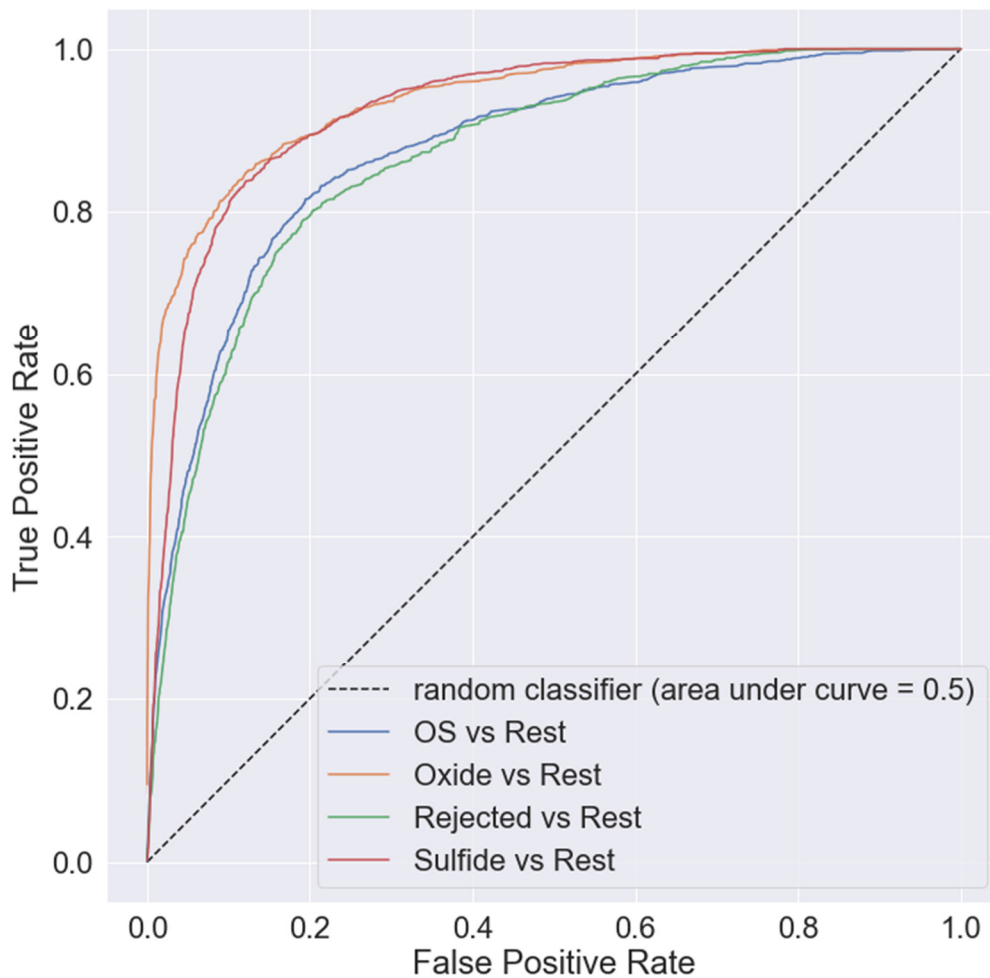


Figure 5-20: ROC curves of Bagging classifier trained on pixel features

5.6 Influence of Inclusion Dimensions

To study the influence of inclusion dimensions on the feature quality and performance of machine learning models, different limits for image sizes were defined for the NMI-class databases. For example, “class database 1: geometric features” was generated several times with different ECD limits. Following ECD values were used for the different NMI-class databases to generate the training sets:

- $\leq 1,5 \mu\text{m}$ \rightarrow 12683 entries (6481 Sulfide, 3458 Rejected, 1487 OS, 1257 Oxide)

- $> 1,5 \mu\text{m}$ and $\leq 2 \mu\text{m}$ \rightarrow 5053 entries (2331 Sulfide, 1678 OS, 610 Rejected, 434 Oxide)
- $> 2 \mu\text{m}$ \rightarrow 8954 entries (5419 OS, 2235 Sulfide, 676 Oxide, 624 Rejected)

To generate balanced datasets across all ECD limitations, alternative sampling was used with a reference number of 1000. Classes in the datasets with a higher number got undersampled, and classes below this threshold oversampled. **Figure 5-21** shows a bar plot with the results of the Bagging classifier trained on the different datasets. It is clearly visible, that the ECD limit and therefore the inclusion dimension influenced the performance. Training on the dataset, which contained NMIs smaller than $1,5 \mu\text{m}$ ECD, led to a significantly smaller accuracy. Depending on the feature type, accuracies only up to 50 % were achievable. Across all features, this behavior was similar. Training on the two datasets containing NMIs with ECD above $1,5 \mu\text{m}$ resulted in accuracy levels ranging from 65 % to 73 %, as previously observed in the dataset without ECD limitations.

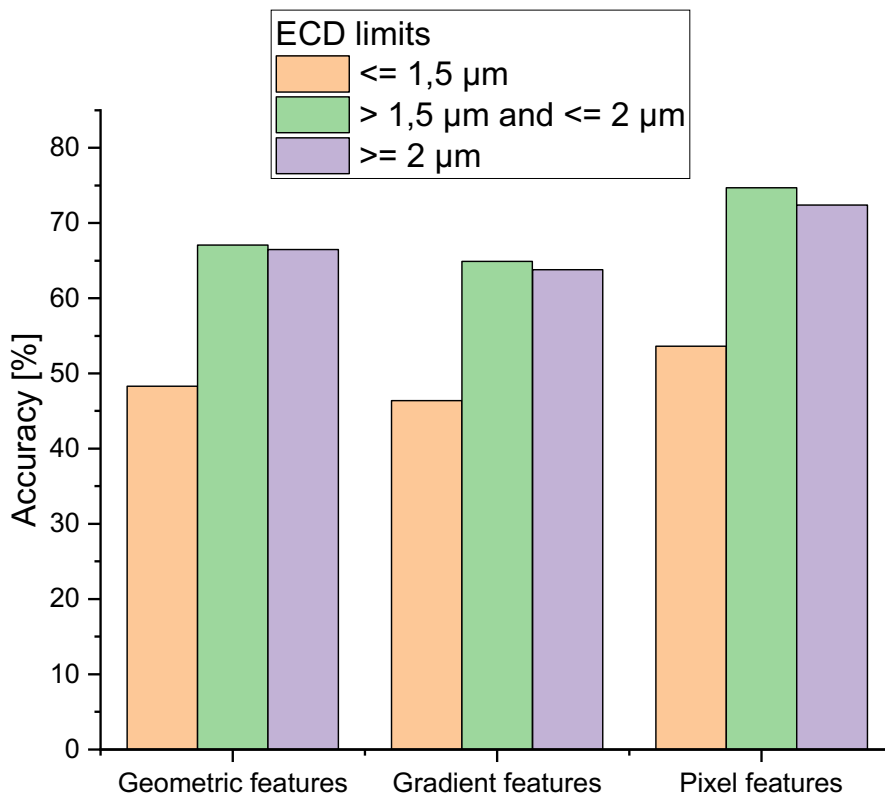


Figure 5-21: Influence of ECD on the accuracy of the Bagging classifier

To remove possible influences from the alternative sampling process on the dataset with NMIs smaller than $1,5 \mu\text{m}$ ECD, a further experiment was done. This trial contained training the Bagging classifier without alternative sampling. The full dataset with the 12683 inclusions was

used. Increasing the amount of data without the balancing technique resulted in an accuracy of 62 % during training with pixel features. Even though this value was higher than previously observed with alternative sampling, a significant gap in performance for the Bagging classifier was still present between the training on inclusions with low and higher ECD.

Considering the amount of training data, the ECD had an even more significant influence on the performance. Training on 12683 NMIs with ECD values smaller than 1,5 μm resulted in 62 % accuracy score, whereas training on 4000 NMIs with ECD values higher than 1,5 μm led to an accuracy over 70 %. Concludingly can be said, that the quality of the extracted features depended on the size of the non-metallic inclusion.

5.7 PyTorch: Deep Learning in Python

This chapter gives a short introduction into deep learning and showcases the application of different neural network types from the PyTorch python library [50] for inclusion characterization. In the literature, several definitions of deep learning exist. Pointer [51] describes deep learning as a machine learning technique, that uses multiple and numerous layers of nonlinear transformations to progressively extract features from the input. Simplified, deep learning is a technique to solve problems by providing the inputs and desired outputs and letting the computer find the solution by using a neural network. A neural network consists of processing units, the neurons, with directed, weighted connections between them. **Figure 5-22** showcases how information travels through a neuron. Propagation functions transform outputs of other neurons to the net input. Activation, defined by the activation function, is the switching status of a neuron. Based on biological models, not every neuron is active at all times. Neurons get activated if the network input exceeds their threshold value. The output function transforms the activation into an output. [51,52]

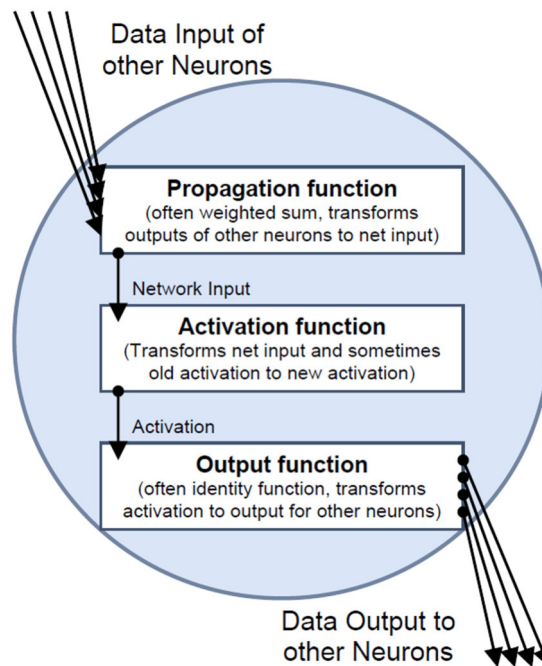


Figure 5-22: Data processing of a neuron [53]

Feedforward networks are deep learning models, where the neurons in one layer have only directed connections to the neurons of the next layer. The neurons are grouped in the input layer, in one or more hidden processing layers, and the output layer. Feedforward networks are very important machine learning tools, as they form the fundament of many deep learning models. The learning algorithm of neural networks is based on the backpropagation algorithm, a supervised learning technique. It involves the following steps:

- Initialization:
 - Random initialization of the weights and biases.
- Forward propagation:
 - Processing of the input in the neural network with the weights and biases to calculate the output.
 - Calculation of the error by comparing the output of the neural network with the true value.
- Backward propagation:
 - Propagation of the error back through the neural network using the chain rule of differentiation.
 - Computation of the gradients for the weights and biases with respect to the error.
- Update weights and biases:

- Updating the values by using the gradients with an optimization algorithm such as stochastic gradient decent in the direction that reduces the error.
- Repeat:
 - Repetition of the steps forward propagation, backward propagation, and update weights and biases until the error falls below a certain threshold.
- Prediction:
 - After training, prediction of the output for new inputs can be fulfilled.

The learning algorithm adjusts the weights and biases of the neural network to minimize the difference between the predicted output and the actual output. By minimizing this difference, the neural network is able to learn to make accurate predictions for new inputs that it has not seen during training. [53,54]

5.7.1 Multilayer Perceptron

A Perceptron is a type of artificial neuron, which represent the basic building block of artificial neural networks. It takes several binary inputs and produces a single binary output. **Figure 5-23** shows an example, where the perceptron processes three inputs, namely x_1 , x_2 , and x_3 , with three weights. [54]

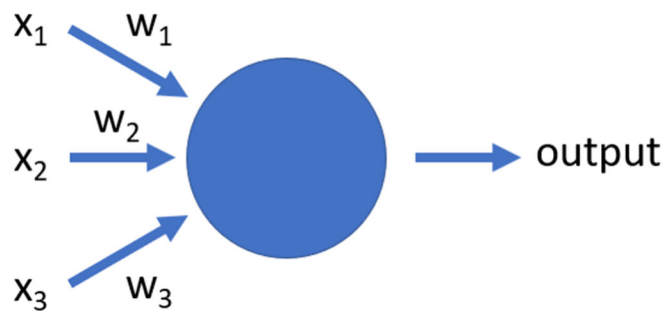


Figure 5-23: Perceptron with three input values

The output gets calculated with the following equation:

$$\text{Output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (5-1)$$

A perceptron with two or more trainable weight layers is called multilayer perceptron. An n-layer perceptron has exactly n variable weight layers and n+1 neuron layers with neuron layer 1 being the input layer. The number of layers, type of activation functions, as well as other

network parameters depend on the type of task which the algorithm needs to solve. Following guidelines were used for the application of multilayer perceptron for inclusion characterization with a 4-class classification [55,56]:

- Data preprocessing:
 - Coding the targets: Four-dimensional targets, in which only one neuron at a time is active (for example (1,0,0,0) represents oxide, (0,1,0,0) represents sulfide, etc.).
- Choice of network architecture:
 - Typically, a neural network starts with a single hidden layer as the standard procedure. In cases where the performance is inadequate, a second hidden layer can be added. Utilizing more than two hidden layers is uncommon.
 - Size of the input layer is defined by the number of features.
 - Determining the number of neurons in the hidden layers depends on the complexity of the function that is being approximated or the decision boundaries that are being implemented. Typically, the approach is to start with more neurons than necessary and then apply early stopping or Bayesian regularization techniques to avoid overfitting.
 - Number of neurons in the last layer is equal to the number of elements in the target vector in a multi output network. Alternatively, it is possible to train four neural networks with only one neuron in the output layer.
 - Transfer function in the output layer: Either sigmoid function or radial basis function.

As a first experiment, a multilayer perceptron with one hidden layer containing 100 neurons was trained on the database with geometric features (split ratio: 60 % training, 20 % validation, 20 % test). Following mentioned performance scores refer to the test dataset. After 100 iterations and a learning rate of 0,01, the accuracy for the 4-class classification approached 56,2 %. Adding a second hidden layer with 100 neurons improved the accuracy score to 57,4 %. Increasing the number of neurons in the hidden layer did not result in a better performance. The same architecture was tested on the database with gradient features, as the number of input variables did not differ significantly compared to the database with geometric features. The accuracy peaked for this experiment at 46,8 %. Regarding pixel features, the architecture of the multilayer perceptron needed to be adapted because 3600 neurons in the input layer were necessary. The first hidden layer contained 1000 neurons and the second 100 neurons. Of all the extracted features, pixel features resulted in the highest accuracy score of 60,7 %. **Figure 5-24** summarizes the results of the multilayer perceptron trained with the

different databases. Especially the recall of oxides for geometric and gradient features showed very low values.

	Accuracy [%]	Precision [%]				Recall [%]			
		OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
Geometric features	56,2	54,6	61,3	45,6	69,6	72,3	15,6	72,9	57,1
Gradient features	46,8	43,1	50,6	36,1	51,0	35,7	23,6	39,9	78,8
Pixel features	60,7	62,8	57,5	50,7	69,0	53,0	66,5	52,8	66,0

Figure 5-24: Results of different multilayer perceptron

Comparing the results of the multilayer perceptron with the Bagging classifier and Random Forest classifier (geometric features: 65,6 %, gradient features: 60,4 %, pixel features: 73,1 %), the deep learning algorithm got outperformed significantly in all three different features.

5.7.2 Convolutional Neural Network

Another type of deep learning algorithm often used for image classification are convolution neural networks (CNN). The basic building blocks of a CNN are convolutional layers, which apply a set of filters, also called kernels, to an input image. During this process, a set of output feature maps are generated. Each filter is a small weight matrix that changes the values during training. Unlike standard neural networks, the neurons within any given layer are only connected to a small region of the layer preceding it. **Figure 5-25** shows a simplified CNN architecture. Starting with a certain input image size, for example 64 x 64 x 3 (height, width, and depth), the dimension gets compromised in the direction to the final output layer by applying convolution and pooling. Pooling layers perform downsampling on the feature maps along the spatial dimensionality to reduce the number of parameters. The final output layer compromises the input to 1 x 1 x n, where n represents the possible number of classes. [57]

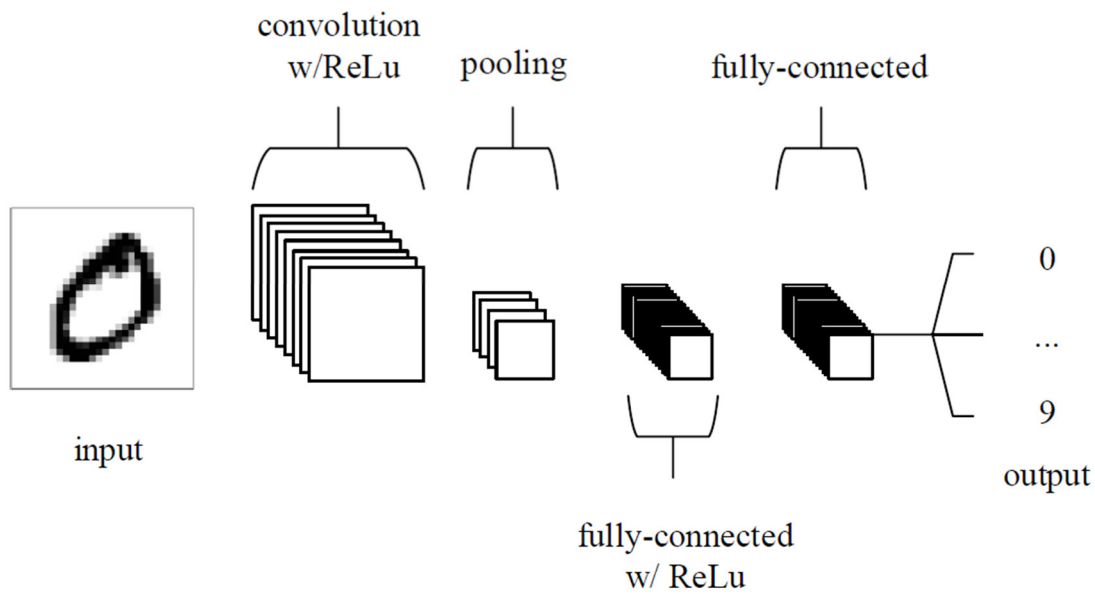


Figure 5-25: CNN architecture [57]

Following implementation guidelines were used for designing the CNN for inclusion characterization [51,57]:

- Common architecture is to use convolutional and pooling layers repeatedly before feeding forward to fully-connected layers.
- Input layer should be recursively dividable by two (for example 32 x 32, or 64 x 64).
- During usage of small filters, stride should be set to one.
- Applying of zero-padding to ensure that convolutional layers do not reconfigure any of the dimensionality.
- Using a dropout layer reduces the tendency to overfit to training data.

As it is shown in **Figure 5-25**, CNNs directly process images. The preprocessing of automated SEM/EDS analysis generated NMI images was done in PyTorch with resizing to an input size of 32 x 32 x 1 using bicubic scaling. The BSE images are grayscale images, hence only one channel exists. After testing different CNN architectures with various hyperparameters, the model presented in **Figure 5-26** achieved the highest accuracy score after 50 training iterations. Overall, the network contained three convolutional layers, three pooling layers, two dropout layers, and two fully connected layers. The activation function 'ReLU' was used after the layers. This network architecture led to approximately 4 million trainable parameters.

Layer (type:depth-idx)	Output Shape
CNNNet	[16, 4]
└Sequential: 1-1	[16, 64, 3, 3]
└Conv2d: 2-1	[16, 16, 32, 32]
└ReLU: 2-2	[16, 16, 32, 32]
└MaxPool2d: 2-3	[16, 16, 15, 15]
└Conv2d: 2-4	[16, 32, 15, 15]
└ReLU: 2-5	[16, 32, 15, 15]
└MaxPool2d: 2-6	[16, 32, 7, 7]
└Conv2d: 2-7	[16, 64, 7, 7]
└ReLU: 2-8	[16, 64, 7, 7]
└MaxPool2d: 2-9	[16, 64, 3, 3]
└AdaptiveAvgPool2d: 1-2	[16, 64, 6, 6]
└Sequential: 1-3	[16, 4]
└Dropout: 2-10	[16, 2304]
└Linear: 2-11	[16, 1152]
└ReLU: 2-12	[16, 1152]
└Dropout: 2-13	[16, 1152]
└Linear: 2-14	[16, 1152]
└ReLU: 2-15	[16, 1152]
└Linear: 2-16	[16, 4]

Figure 5-26: Best performing CNN architecture

The learning rate was set to a value of 0,001. After training the CNN, performance metrics were calculated for the test set (Figure 5-27). The highest error occurred for the classification of oxides, showing a low recall of 22,3 %.

Accuracy [%]	Precision [%]				Recall [%]			
	OS	Oxide	Rejected	Sulfide	OS	Oxide	Rejected	Sulfide
67,2	71,0	46,0	59,0	75,0	60,2	22,3	79,3	75,4

Figure 5-27: Performance metrics of the described CNN architecture

To showcase, in which classes the CNN distinguished the oxides, a confusion chart is presented in Figure 5-28. The pronounced oxide/rejected misclassification resulted in the low oxide recall score, as almost 60 % of the oxides got wrongly classified as rejected particles.

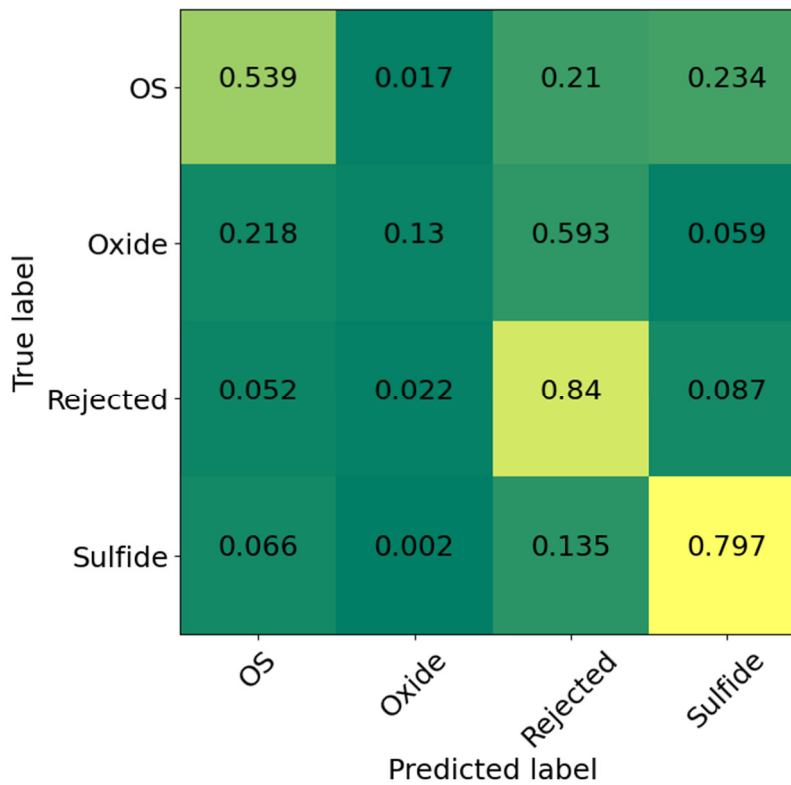


Figure 5-28: Confusion matrix of the trained CNN

6 Conclusion and Outlook

In this work, various databases and machine learning algorithms were tested and evaluated for the application as an inclusion characterization tool. Four different types of features, namely number of pixels with a gray value of 9, geometric, gradient, and pixel features, were extracted

from the NMIs and used as basis for an NMI-class classification ('oxide', 'OS', 'rejected', 'sulfide') and an NMI-type classification ('(Mn)S', 'not typified', '(Mn)O<S', '(Mn,Ca)S', '(Mn,Ti)S', '(Al,Ca,Mn)OS', 'SiC/matrix').

Regarding NMI-class classification, the machine learning algorithms performed the best on pixel features, achieving an accuracy score of 73,1 % after fine-tuning the Random Forest classifier. Classification with geometric features was significantly worse, which resulted in an accuracy of 65,6 %. With gradient features, the algorithms achieved 60 %, and using the number of pixels with gray value of 9 as a distinguishing factor between the classes did not deliver satisfying results. The NMI dimensions had a significant impact on the performance of the classifier. Due to the low information content in BSE images of small inclusions with a ECD of 1 μm , difficulties for the classification occurred. Only pixel features could be used for NMIs with an ECD of 1 μm , as other input variables relied on a certain image size to ensure a good representation quality.

A comparison between the results of NMI-class and NMI-type classification leads to the conclusion, that both approaches resulted in similar performance levels if the inherently greater difficulty of the NMI-type classification is considered. This work focused mainly on the evaluation of NMI-class databases because more data was available in the NMI-classes rather than in NMI-types.

Bagging classifier and Random Forest classifier performed on similar accuracy levels across the different databases. The application of multilayer perceptron resulted in significantly lower accuracy scores, approximately by 15 %. Convolutional neural network achieved 67,2 % accuracy, but problems with oxide/rejected misclassifications occurred. Regarding the performance of all machine learning models can be concludingly stated, that the type of feature used for classification, the type of algorithm, as well as the sampling technique influenced the final result significantly.

Important for further studies on this field is the consideration of the Feature Evaluation Tool with the standardized classification criteria using element thresholds of 0,1 wt.%. Low limits for classification are suitable with EDS data, but elements with this low mass percentage may not show a significant impact on geometric or pixel features for machine learning. Furthermore, standard SEM settings were used to generate the databases. As showcased by Ramesh Babu [27], performance of classifier can be increased by adapting the contrast and brightness values during calibration of the automated SEM/EDS measurement.

For a real-world application, a lot of different influencing factors on the data need to be considered and adapted accordingly to produce a stable machine learning model. In this work,

the influence of different features on various algorithms was showcased. Furthermore, knowledge about metallurgical parameters (influence the size of NMI, inclusion population, etc.), sample preparation steps, SEM measurement parameters (influence the feature quality), and algorithm parameters is essential to minimize the possibility of a domain shift during application.

Bibliography

- [1] T. Wuest, D. Weimer, C. Irgens and K.-D. Thoben, Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research* 4 (2016), 1, pp. 23–45. doi:10.1080/21693277.2016.1192517
- [2] J. Brandenburger, V. Colla, G. Nastasi, F. Ferro, C. Schirm and J. Melcher, Big Data Solution for Quality Monitoring and Improvement on Flat Steel Production. *IFAC-PapersOnLine* 49 (2016), 20, pp. 55–60. doi:10.1016/j.ifacol.2016.10.096
- [3] S. Guo, J. Yu, X. Liu, C. Wang and Q. Jiang, A predicting model for properties of steel using the industrial big data based on machine learning. *Computational Materials Science* 160 (2019), pp. 95–104. doi:10.1016/j.commat.2018.12.056
- [4] C. Wang, C. Shen, Q. Cui, C. Zhang and W. Xu, Tensile property prediction by feature engineering guided machine learning in reduced activation ferritic/martensitic steels. *Journal of Nuclear Materials* 529 (2020), pp. 151823. doi:10.1016/j.jnucmat.2019.151823
- [5] L. Qiao, J. Zhu and Y. Wang, Machine Learning-Aided Process Design: Modeling and Prediction of Transformation Temperature for Pearlitic Steel. *steel research int.* 93 (2022), 1, pp. 2100267. doi:10.1002/srin.202100267
- [6] H. Mesa, D. Urda, J.J. Ruiz-Aguilar, J.A. Moscoso-López, J. Almagro, P. Acosta and I.J. Turias, A Machine Learning Approach to Determine Abundance of Inclusions in Stainless Steel, in: H. Pérez García, L. Sánchez González, M. Castejón Limas, H. Quintián Pardo, E. Corchado Rodríguez (Eds.), *Hybrid Artificial Intelligent Systems*. pp. 504–513, Cham (2019), Springer International Publishing.
- [7] Z. Sun and K. Wang, A screening strategy for hot forging combining high-throughput forging experiment and machine learning. *Mater. Res. Express* 7 (2020), 11, pp. 116509. doi:10.1088/2053-1591/abc4f7
- [8] T.L. Wu, Y.C. Hwang and W.X. Zhang, Machine learning-based model for detecting uneven wear and temperature deviation events in hot forging process. *International Journal of Advanced Manufacturing Technology* 119 (2022), 3-4, pp. 2743–2761. doi:10.1007/s00170-021-08256-z

- [9] A. Mayerhofer, Enhanced Characterization of Non-Metallic Inclusions for (Sub) Micro Steel Cleanliness Evaluations. Dissertation, Leoben (2021).
- [10] H. C., The Importance of the Non-Metallic Inclusions in Steel. *Nature* 101 (1918), 2539, pp. 334–335. doi:10.1038/101334a0
- [11] L. Zhang and B.G. Thomas, State of the Art in Evaluation and Control of Steel Cleanliness. *ISIJ International* 43 (2003), 3, pp. 271–291. doi:10.2355/isijinternational.43.271
- [12] A.L.V. Da Costa e Silva, Non-metallic inclusions in steels – origin and control. *Journal of Materials Research and Technology* 7 (2018), 3, pp. 283–299. doi:10.1016/j.jmrt.2018.04.003
- [13] J. Burja, M. Koležnik, Š. Župerl and G. Klančnik, Nitrogen and nitride non-metallic inclusions in steel. *Mater. Tehnol.* 53 (2019), 6, pp. 919–928. doi:10.17222/mit.2019.247
- [14] B.H. Reis, W.V. Bielefeldt and A.C.F. Vilela, Absorption of non-metallic inclusions by steelmaking slags—a review. *Journal of Materials Research and Technology* 3 (2014), 2, pp. 179–185. doi:10.1016/j.jmrt.2014.03.011
- [15] N. Cyril and A. Fatemi, Experimental evaluation and modeling of sulfur content and anisotropy of sulfide inclusions on fatigue behavior of steels. *International Journal of Fatigue* 31 (2009), 3, pp. 526–537. doi:10.1016/j.ijfatigue.2008.04.001
- [16] P. Kaushik, H. Piolet and H. Yin, Inclusion characterisation – tool for measurement of steel cleanliness and process control: Part 2. *Ironmaking & Steelmaking* 36 (2009), 8, pp. 572–582. doi:10.1179/030192309X12492910938177
- [17] S.K. Michelic and C. Bernhard, Significance of Nonmetallic Inclusions for the Clogging Phenomenon in Continuous Casting of Steel—A Review. *steel research int.* 93 (2022), 7, pp. 2200086. doi:10.1002/srin.202200086
- [18] J.M. Wang, Y. Zhang, Y.F. Wang and X.Q. Chen, Influence of Inclusions in Low Alloy-Steels on Pitting Process. *AMR* 396-398 (2011), pp. 395–399. doi:10.4028/www.scientific.net/AMR.396-398.395
- [19] J. Zhang, S. Li, Z. Yang, G. Li, W. Hui and Y. Weng, Influence of inclusion size on fatigue behavior of high strength steels in the gigacycle fatigue regime. *International Journal of Fatigue* 29 (2007), 4, pp. 765–771. doi:10.1016/j.ijfatigue.2006.06.004
- [20] K. Yamamoto, H. Yamamura and Y. Suwa, Behavior of Non-metallic Inclusions in Steel during Hot Deformation and the Effects of Deformed Inclusions on Local Ductility. *ISIJ International* 51 (2011), 12, pp. 1987–1994. doi:10.2355/isijinternational.51.1987

- [21] P. Kaushik, H. Pielet and H. Yin, Inclusion characterisation – tool for measurement of steel cleanliness and process control: Part 1. *Ironmaking & Steelmaking* 36 (2009), 8, pp. 561–571. doi:10.1179/030192309X12492910938131
- [22] J. Ruuska, S. Ollila and K. Leiviskä, The Possibility to Use Optical Emission Spectrometry for Identifying the Amount of Inclusions in Steels. *MSF* 762 (2013), pp. 649–655. doi:10.4028/www.scientific.net/MSF.762.649
- [23] Zacharias Klußmann, Untersuchungen zur Fehlerquellenidentifikation an Stahlproben durch Nutzung des SILENOS-Prüfsystems. Masterarbeit, Leoben (2016).
- [24] B. Bandi, B. Santillana, W. Tiekink, N. Koura, M. Williams and P. Srirangam, 2D automated SEM and 3D X-ray computed tomography study on inclusion analysis of steels. *Ironmaking & Steelmaking* 47 (2020), 1, pp. 47–50. doi:10.1080/03019233.2019.1652437
- [25] L. Reimer, *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis*. 45. Auflage (1998).
- [26] M. Abdulsalam, T. Zhang, J. Tan and B.A. Webler, Automated Classification and Analysis of Non-metallic Inclusion Data Sets. *Metall Mater Trans B* 49 (2018), 4, pp. 1568–1579. doi:10.1007/s11663-018-1276-x
- [27] S. Ramesh Babu, R. Musi, K. Thiele and S.K. Michelic, Classification of Nonmetallic Inclusions in Steel by Data-Driven Machine Learning Methods. *steel research int.* 94 (2023), 1, pp. 2200617. doi:10.1002/srin.202200617
- [28] M. Abdulsalam, M. Jacobs and B.A. Webler, Automated Detection of Non-metallic Inclusion Clusters in Aluminum-deoxidized Steel. *Metall Mater Trans B* 52 (2021), 6, pp. 3970–3985. doi:10.1007/s11663-021-02312-5
- [29] R. Musi, Klassifikation von nichtmetallischen Einschlüssen mittels deep learning Algorithmen. Bachelorarbeit, Leoben (2021).
- [30] M. Abdulsalam, N. Gao, B.A. Webler and E.A. Holm, Prediction of Inclusion Types From BSE Images: RF vs. CNN. *Front. Mater.* 8 (2021). doi:10.3389/fmats.2021.754089
- [31] S.L. Brunton and J.N. Kutz, *Data-Driven Science and Engineering* (2019), Cambridge University Press.
- [32] R. Wirth and J. Hipp, *Crisp-dm: towards a standard process modell for data mining* (2000).
- [33] G. Chandrashekar and F. Sahin, A survey on feature selection methods. *Computers & Electrical Engineering* 40 (2014), 1, pp. 16–28. doi:10.1016/j.compeleceng.2013.11.024

- [34] T. Duboudin, E. Dellandréa, C. Abgrall, G. Hénaff and L. Chen, Encouraging Intra-Class Diversity Through a Reverse Contrastive Loss for Better Single-Source Domain Generalization. ICCV - Workshop on Adversarial Robustness In the Real World. doi:10.48550/arXiv.2106.07916
- [35] M. Gadermayr, A. Uhl and A. Vécsei, Dealing with Intra-Class and Intra-Image Variations in Automatic Celiac Disease Diagnosis, in: H. Handels (Ed.), *Bildverarbeitung für die Medizin 2015*. pp. 461–466, Berlin, Heidelberg (2015), Springer Berlin Heidelberg.
- [36] Theophano Mitsa, How Do You Know You Have Enough Training Data? <https://towardsdatascience.com/how-do-you-know-you-have-enough-training-data-ad9b1fd679ee>, Accessed: 25.03.2023
- [37] R. Szeliski, *Computer Vision: Algorithms and Applications* (2010), Springer.
- [38] B. Goyal, A. Dogra, S. Agrawal and B.S. Sohi, Noise Issues Prevailing in Various Types of Medical Images. *Biomed. Pharmacol. J.* 11 (2018), 3, pp. 1227–1237. doi:10.13005/bpj/1484
- [39] A. Piovesana and G. Senior, How Small Is Big: Sample Size and Skewness. *Assessment* 25 (2018), 6, pp. 793–800. doi:10.1177/1073191116669784
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [41] O. Chapelle, Training a support vector machine in the primal. *Neural computation* 19 (2007), 5, pp. 1155–1178. doi:10.1162/neco.2007.19.5.1155
- [42] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. 14. Auflage, Cambridge (2014), Cambridge University Press.
- [43] R. Mohammed, J. Rawashdeh and M. Abdullah, Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 243–248, Irbid, Jordan (2020), IEEE.
- [44] A. GERON, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools and techniques to build intelligent systems*. 2nd ed., Beijing, Boston, Farnham [etc.] (2019), O'Reilly.
- [45] javaTpoint, Decision Tree Classification Algorithm. <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>, Accessed: 01.04.2023

- [46] G. Louppe and P. Geurts, Ensembles on Random Patches, in: D. Hutchison, T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell et al. (Eds.), *Machine Learning and Knowledge Discovery in Databases*. pp. 346–361, Berlin, Heidelberg (2012), Springer Nature.
- [47] scikit-learn, Ensemble Methods. <https://scikit-learn.org/stable/modules/ensemble.html#bagging>, Accessed: 01.04.2023
- [48] scikit-learn, Multiclass Receiver Operating Characteristics (ROC). https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html, Accessed: 01.04.2023
- [49] Wikipedia, Receiver operating characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic, Accessed: 01.04.2023
- [50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: *Advances in Neural Information Processing Systems 32*. pp. 8024–8035 (2019), Curran Associates, Inc.
- [51] I. Pointer, *Programming PyTorch for deep learning creating and deploying deep learning applications* (2019), O'Reilly.
- [52] S.S. Haykin, *Neural networks and learning machines*. 3rd edition, Upper Saddle River, N.J. (2009), Prentice Hall.
- [53] D. Kriesel, *A Brief Introduction to Neural Networks*. <http://www.dkriesel.com>, Accessed: 02.04.2023
- [54] M. Nielsen, *Using neural nets to recognize handwritten digits*. <http://neuralnetworksanddeeplearning.com/chap1.html>, Accessed: 10.04.2023
- [55] M.T. Hagan, *Neural network design*. <https://hagan.okstate.edu/NNDesign.pdf>, Accessed: 14.04.2023
- [56] J. Heaton, *Introduction to Neural Networks for Java, Second Edition*. St. Louis, Mo. (2008), Heaton research.
- [57] K. O'Shea and R. Nash, *An Introduction to Convolutional Neural Networks*. CoRR [abs/1511.08458](https://arxiv.org/abs/1511.08458) (2015). doi:10.48550/arXiv.1511.08458

Acronyms

BSE	Backscattered electrons
CNN	Convolutional Neural Network
CRISP-DM	Standard Cross-Industry for Process Data Mining
CT	Computed tomography
ECD	Equivalent circle diameter
FET	Feature Evaluation Tool
MIDAS	Mannesmann Inclusion Detection by Analyzing Surfboards
ML	Machine learning
N	Nitride
NMI	Non-metallic inclusion
NS	Nitride-sulfide
O	Oxide
OES-PDA	Optical emission spectrometry with pulse discrimination analysis
OM	Optical microscope
ON	Oxide-nitride
ONS	Oxide-nitride-sulfide
OOB	Out-of-bag
OS	Oxide-sulfide
PC	Principal Component
PCA	Principal Component Analysis
QT	Quenched and tempered
RF	Random Forest
ROC	Receiver operating characteristics
S	Sulfide
SEM/EDS	Scanning electron microscope with energy dispersive spectroscopy
SGD	Stochastic Gradient Decent
SVM	Support Vector Machine
US	Ultrasonic testing
WDS	Wavelength-dispersive

List of Tables

Table 2-I: Artifact correction criteria.....	9
Table 2-II: Classification thresholds.....	9
Table 4-I: Measurement parameter for field-emitter-based JEOL 7200F	20
Table 4-II: Comparison of oxide, sulfide, and oxide-sulfide inclusions in high- and low-resolution images	27
Table 4-III: Comparing the results of the gray value gradients of oxide, sulfides, and OS.....	31
Table 4-IV: Comparison of gray value distributions of the most important classes from the dataset	38
Table 5-I: Classifier models from scikit-learn library	42
Table 5-II: Hyperparameter settings for Random Forest classifier	55

List of Figures

Figure 2-1: Schematic representation of an inclusion approaching and breaking through the steel/slag interface [12].....	4
Figure 2-2: Comparison of the most common direct methods for steel cleanliness evaluation with measuring limits and characteristics [9].....	6
Figure 2-3: Gray value calibration of a multi-phase NMI for automated SEM/EDS analysis	8
Figure 2-4: Further classification of multi-phase inclusions based on the share of the non-metallic phase [9]	10
Figure 3-1: PCA scatter plots of four samples from a 4140 steel [26].....	12
Figure 3-2: Comparison of observed and predicted values of the average abundance of inclusions from a linear regression model [6].....	13
Figure 3-3: Schematic overview of different data approaches for the application of machine learning for inclusion characterization.....	14
Figure 3-4: Influence of contrast on the gray value of oxide and sulfide in oxide-sulfide inclusions [27]	15
Figure 3-5: Confusion matrix of an VGG16 network trained on BSE-images [29].....	16
Figure 3-6: Multiphase inclusions with wrong determined chemical composition [30].....	17
Figure 4-1: Flowchart of how data, starting from Aztec software, is processed (color program environments - black: Aztec, green: FET, yellow: Windows, blue: Python)	19
Figure 4-2: Number of NMIs in each class for the different steel samples.....	22
Figure 4-3: Number of NMIs per mm ² in each class for the different steel samples.....	23
Figure 4-4: Number of NMIs in each class for the whole dataset	24
Figure 4-5: Oxide inclusion types in the QT steel.....	25
Figure 4-6: Amount of the twenty most common types in the dataset.....	26
Figure 4-7: Influence of ECD on mean gray level of oxide inclusions in the QT steel	28

Figure 4-8: Low-resolution BSE images of small oxide inclusions (ECD $\approx 1,5 \mu\text{m}$)	28
Figure 4-9: Gray value distribution of every image in the dataset.....	30
Figure 4-10: Number of pixels with a gray value of 9 from NMI classes in the dataset.....	30
Figure 4-11: Summed up gray value distribution of every image in the respective steel grades (excluding the austenitic steel due to low number of sulfides).....	32
Figure 4-12: ECD distribution of the dataset	33
Figure 4-13: Relative share of images with a certain mean gray value.....	34
Figure 4-14: Median, mean, 25 % and 75 % quantiles of the mean gray values of all images	35
Figure 4-15: Relative share of images with a certain median gray value.....	36
Figure 4-16: Correlation matrix of geometric parameters	37
Figure 4-17: Sulfide-type distribution of the steels.....	40
Figure 5-1: Class distribution of the NMI-class databases.....	41
Figure 5-2: Type distribution of the NMI-type databases	42
Figure 5-3: Under- and oversampling of a dataset [43]	44
Figure 5-4: Accuracy of classifiers for under- and oversampling	45
Figure 5-5: Precision and recall of Bagging classifiers trained on over- and undersampled NMI-class databases	46
Figure 5-6: Precision and recall of Random Forest classifiers trained on over- and undersampled NMI-class databases.....	46
Figure 5-7: Result of the alternative sampling strategy on the NMI-type distribution	47
Figure 5-8: Comparison between alternative-, under-, and oversampling in the NMI-class databases with the performance of Bagging classifier	48
Figure 5-9: Comparison between class- and type-labelling	49
Figure 5-10: Sampling and training process of Bagging classifiers [44].....	50
Figure 5-11: Schematic representation of a Decision Tree [45].....	50
Figure 5-12: Precision and recall for BC and RF trained on alternative sampled NMI-class databases.....	51

Figure 5-13: Correlation between accuracy and number of estimators for the Bagging classifier	52
Figure 5-14: Result of using the Random Patches Method on pixel features with randomized search	53
Figure 5-15: Achieved performance increase with fine-tuning the Bagging classifier's hyperparameter	54
Figure 5-16: Achieved performance increase with fine-tuning the RF classifier's hyperparameter	55
Figure 5-17: Feature importance of pixel from BSE images	57
Figure 5-18: Exemplary ROC curve [49]	57
Figure 5-19: ROC curves of Bagging classifier trained on geometric features.....	58
Figure 5-20: ROC curves of Bagging classifier trained on pixel features.....	59
Figure 5-21: Influence of ECD on the accuracy of the Bagging classifier	60
Figure 5-22: Data processing of a neuron [53]	62
Figure 5-23: Perceptron with three input values	63
Figure 5-24: Results of different multilayer perceptron.....	65
Figure 5-25: CNN architecture [57]	66
Figure 5-26: Best performing CNN architecture	67
Figure 5-27: Performance metrics of the described CNN architecture.....	67
Figure 5-28: Confusion matrix of the trained CNN.....	68

A Appendix

A.1 Chemical Compositions of the Steels

Following chemical compositions were measured with a spark spectrometer at the Chair of Ferrous Metallurgy.

Table A-1: Chemical compositions of the steels from the dataset (wt%)

Steel	C	Si	Mn	P	S	Cr	Mo	Ni	Al	Nb	Fe
Quenched and tempered steel	0,450	0,303	0,743	0,0065	0,0036	1,14	0,201	0,224	0,0167	< 0,001	Bal.
Austenitic steel	0,0226	0,252	1,85	0,0237	< 0,001	17,13	2,73	14,91	0,0025	< 0,001	Bal.
Construction steel	0,431	0,254	0,804	0,013	0,0304	0,979	0,201	0,0382	0,0275	< 0,001	Bal.
Rail steel	0,789	0,421	1,1	0,0158	0,0174	0,0604	0,0069	0,0207	< 0,001	< 0,001	Bal.
Spring steel	0,524	0,299	1,03	0,0143	0,0186	1,11	0,0109	0,0432	0,0229	< 0,001	Bal.
Micro alloyed steel	0,0765	< 0,001	0,936	0,0142	0,007	0,029	0,0018	0,0332	0,0293	0,0312	Bal.
Bearing steel	0,166	0,27	1,41	0,0116	0,0032	0,679	0,46	1,02	0,0327	0,0377	Bal.