FACULTY OF ENGINEERING
AND ARCHITECTURE

Machine Learning for Microstructure Analysis of Steel

Michiel Larmuseau

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors

Prof. Stefaan Cottenier, PhD* - Prof. Tom Dhaene, PhD** - Lode Duprez, PhD***

* Department of Electromechanical, Systems and Metal Engineering
Faculty of Engineering and Architecture, Ghent University

** Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

***OCAS

May 2021

GHENT
UNIVERSITY

ISBN 978-94-6355-483-1
NUR 984, 971
Wettelijk depot: D/2021/10.500/31

Members of the Examination Board

Chair

Prof. Em. Daniël De Zutter, PhD, Ghent University

Other members entitled to vote

Prof. Elizabeth Holm, PhD, Carnegie Mellon University, USA
Prof. Leo Kestens, PhD, Ghent University
Prof. Aleksandra Pizurica, PhD, Ghent University
Prof. Dirk Roose, PhD, KU Leuven
Koenraad Theuwissen, PhD, OCAS

Supervisors

Prof. Stefaan Cottenier, PhD, Ghent University
Prof. Tom Dhaene, PhD, Ghent University
Lode Duprez, PhD, OCAS

Preface

Hoewel de resultaten die ik in deze thesis bespreek voornamelijk het gevolg zijn van mijn eigen noeste arbeid, zijn er een aantal mensen die tijdens mijn doctoraat een belangrijke rol gespeeld hebben. Het lijkt mij gepast om hen hier te bedanken.
Allereerst wens ik mijn promotoren Lode, Stefaan en Tom te bedanken om mij enerzijds de wetenschappelijke vrijheid en het vertrouwen te geven om mijn eigen weg te zoeken in de wondere wereld van het academisch onderzoek en om mij anderzijds te helpen op de momenten dat het nodig was.
Daarnaast wil ik Koenraad, Kurt en Michaël bedanken voor hun input en de hulp bij het schrijven van de publicaties. Koenraad moet ik ook bedanken voor het organiseren van de quizzen en de vele hulp bij het analyseren van de microstructuren.
Mijn kantoorgenoten Merlijn, Michaël en Sam wil ik bedanken voor de vele gezellige momenten samen. De sfeer zat altijd goed op kantoor en in zo een omgeving voelde mijn doctoraat niet aan als werken. Daarnaast wil ik mijn collega's op het CMM bedanken voor de boeiende en soms absurde gesprekken in de keuken. Klaas en Arthur wens ik expliciet te bedanken voor de vele raadseltjes en wiskundige problemen die zij mij voorgeschoteld hebben. Van SUMOLab wil ik Baptist, Ivo, Leen en Tom bedanken voor de inhoudelijke gesprekken die we gehad hebben en die mij veel geholpen hebben om richting te geven aan mijn onderzoek. Ik wil ook de mensen van OCAS bedanken voor hun input en enthousiasme tijdens de meetings. Dit heeft mij veel geholpen om steeds op relevante problemen te focussen.
Tenslotte wil ik mijn familie en vrienden bedanken om een luisterend oor te bieden op de momenten dat ik mijn hart eens moest luchten. In het bijzonder moet ik mijn ouders, mijn broer en mijn zus bedanken om het meerdere lockdowns met mij vol te hebben gehouden. Maarten moet ik ook bedanken voor het maken van Figuur 3.2 (b) in deze thesis.

Michiel Larmuseau
Gent, 8 juni 2021

Contents

List of Symbols

Alphanumeric symbols	
F	applied load
A	surface area
A_0	initial cross-section
D	diagonal of indenter ball
d	diagonal of indentation
L	length
L_0	initial length
L_f	final length
X	image matrix
H	image height
W	image width
P'	non-normalised GLCM
P	normalised GLCM
P_x	marginal GCLM of x
P_y	marginal GCLM of y
H_{xy}	entropy of the joint distribution of x and y
H_x	entropy of the marginal distribution of x
H_y	entropy of the marginal distribution of y
n	a local microstructural state
$\mathbf{r}$	space vector
m_s^h	microstructure function at location s for local state h
$f_s^{hh'}$	two-point correlation function at location s for local states h and h'
$\bar{f}_s$	average of the two-point correlation function at location s
$\mathbf{x}$	input vector
W	weight matrix

$\mathbf{b}$	bias vector
$\mathbf{y}$	output vector
C	number of image channels
S	kernel size
p	dropout probability
r	spatial resolution increase factor
f_θ	deep neural network with trainable parameters $\boldsymbol{\theta}$
L	loss function
$\boldsymbol{i}_l$	input of layer i
$\boldsymbol{o}_l$	output of layer i
$\boldsymbol{v}_t$	velocity or second moment of the gradient at iteration t depending on the context
$\boldsymbol{m}_t$	first moment of the gradient at iteration t
$\hat{\boldsymbol{v}}_t$	unbiased second moment of the gradient at iteration t
$\hat{\boldsymbol{m}}_t$	unbiased first moment of the gradient at iteration t
n_l	number of input or output features of layer l
C_{43}	penultimate convolutional block in a VGG16 network
f_*	new prediction of Gaussian process
I	identity matrix
x_*	new input sample
K	covariance kernel
l_i	lengthscale of feature i

Greek symbols

σ	stress or standard deviation depending on context
σ_x	standard deviation of the marginal distribution of x
σ_y	standard deviation of the marginal distribution of y
σ_B	batch standard deviation
ϵ	strain or small value used for numerical stability depending on context
μ	mean of the distribution or momentum parameters depending on the context
μ_x	mean of the marginal distribution of x
μ_y	mean of the marginal distribution of y
μ_B	batch mean
ϕ_{is}	i-th principal component at location s
$\alpha_i^{hh'}$	i-th coefficient of in PCA basis for local states h and h'
$\boldsymbol{\gamma}$	trainable scale parameters in batch-normalization layers
$\boldsymbol{\beta}$	trainable offset parameters in batch-normalization layers
$\boldsymbol{\theta}$	trainable model parameters

$\boldsymbol{\theta}_l$	trainable parameters of layer l
$\boldsymbol{\theta}_t$	value of the trainable parameters at iteration t
$\boldsymbol{\delta}_l$	derivative of the loss to the output of layer l
η	learning rate
β_1	decay rate for the first moment of the gradient
β_2	decay rate for the second moment of the gradient
$\mathcal{N}$	normal distribution
α	margin in the triplet loss

In general, vectors are printed in bold face throughout this work, whereas matrices and tensors are printed in capital letters.

List of Abbreviations

CFRPs	Carbon Fibre Reinforced Polymers
PSP	processing structure properties
wt. %	percentage by weight
HV	Vickers hardness
HB	Brinell hardness
%EL	percentage elongation
CCT	continuous cooling transformation
OM	optical microscopy
SEM	scanning electron microscopy
DIC	differential interference contrast
FEG	field emission gun
GLCM	grey-level co-occurrence matrix
LBP	local binary pattern
HoG	histogram of oriented gradients
TAS	threshold adjacency statistic
PCA	principal component analysis
ILSVRC	ImageNet large scale visual recognition competition
ReLU	rectifying linear unit
ResNet	residual network
acc.	accuracy
CNN	convolutional neural network
SISR	single image super-resolution
GAN	generative adversarial network
MAE	mean absolute error
bcc	body-centered cubic
fcc	face-centered cubic

Samenvatting

De ontwikkeling van nieuwe materialen is altijd al een belangrijke motor van technologische vooruitgang geweest. In de Bronstijd hadden de beschavingen die brons konden maken een duidelijk technologisch voordeel ten opzichte van andere beschavingen. Later, in de Ijzertijd, was dit opnieuw het geval. Tot op de dag van vandaag speelt ijzer een belangrijke rol in onze maatschappij. Vooral staal, ijzer met een kleine hoeveelheid koolstof, wordt in verschillende industrieën gebruikt zoals de bouwindustrie, de werktuigindustrie en de automobielindustrie. Jaarlijks wordt er wereldwijd meer dan 1 800 miljoen ton staal geproduceerd. Omdat staal zo intensief gebruikt wordt, kan een kleine verbetering in de eigenschappen ervan leiden tot enorme kostenbesparingen. Om de eigenschappen van staal op een systematische manier te kunnen verbeteren, is een goed begrip van de relatie tussen de behandeling, de structuur en de eigenschappen van het materiaal onontbeerlijk. De structuur van het staal op kleine schaal, ook wel de microstructuur genaamd, wordt typisch bestudeerd met behulp van microscopieafbeeldingen. Het is echter niet eenvoudig om hetgeen men op zo een afbeelding ziet te linken aan de behandeling of de eigenschappen van het materiaal. Metallurgen zijn daarom al decennialang op zoek naar methodes om de microstructuur samen te vatten in een beperkt aantal getallen, zodat de microstructuur op een meer kwantitatieve manier in relatie gebracht kan worden met zowel de behandeling als de eigenschappen van het materiaal.

In de afgelopen jaren hebben deep learning methodes zich ontpopt tot een onontbeerlijk instrument voor de analyse van afbeeldingen in verschillende disciplines zoals geneeskunde, biologie en astronomie. Eén van de redenen waarom deep learning modellen zo goed zijn in het analyseren van afbeeldingen is dat ze uit zichzelf en op basis van de beschikbare afbeeldingen kunnen leren welke kenmerken van afbeelding belangrijk zijn. Vroegere machine learning methodes vereisten immers dat men op basis van domeinkennis zelf de belangrijkste kenmerken uit de afbeeldingen ging halen, wat heel tijdrovend kon zijn. Een intrinsiek nadeel van deep learning is echter dat het veel data nodig heeft om goede modellen op te leveren.

Het doel van dit doctoraat is te onderzoeken welke rol machine learning en vooral deep learning kan spelen in de analyse van microscopieafbeeldingen van microstructuren. We beginnen met te onderzoeken hoe deep learning kan aangewend worden om een microstructuren te beschrijven in een beperkt aantal getallen. We bestuderen hiervoor een methode genaamd triplet netwerken. Deze methode leert microstructuren zo voor te stellen dat sterk gelijkaardige materialen voorstellingen zullen hebben die dicht bij elkaar liggen, terwijl visueel verschillende materialen voorstellingen zullen hebben die ver van elkaar liggen. Met andere woorden, de afstand is een maat voor hoe visueel gelijkaardig verschillende afbeeldingen zijn. Wanneer we deze methode toepassen op een grote dataset met afbeeldingen van zestig verschillende materialen, vinden we dat door de microstructuur samen te vatten in slechts twee getallen, we reeds in staat zijn om sterk gelijkaardige materialen uit elkaar te houden. Als we meer getallen gebruiken om de microstructuur te beschrijven, wordt het nog gemakkelijker om de verschillende materialen uit elkaar te houden. We besluiten daarom dat triplet netwerken metallurgen kunnen helpen om microstructuren te beschrijven. Wanneer we echter de beschrijving voor een specifieke set van materialen proberen toe te passen op een nieuwe set materialen, merken we dat deze veel minder informatie bevat. We vermoeden dat dit kan opgelost worden door grotere datasets met een grotere verscheidenheid in zowel materialen als microstructuren te gebruiken.

De vaststelling dat deep learning modellen goed in staat blijken om verschillen te zien tussen sterk gelijkaardige microstructuren -iets waar zelfs doorwinterde metallurgen problemen mee hebben- werpt een interessante vraag op: kunnen deep learning modellen microstructuren beter herkennen dan ervaren metallurgen? Om dit nader te onderzoeken, hebben we twee quizzen georganiseerd waarin zowel experts als een deep learning model een aantal afbeeldingen moest toekennen aan een vooraf bepaalde set materialen. De eerste quiz bestond uit materialen die duidelijk verschillend waren, terwijl de tweede quiz op een specifieke soort microstructuren focuste. In beide quizzen behaalde het deep learning model een hogere score dan de gemiddelde expert en in de tweede quiz behaalde het model zelfs een perfecte score. Dit is des te opmerkelijker omdat het model slechts een vijftal afbeeldingen per materiaalklasse gezien had tijdens de training. Onze resultaten leiden tot twee belangrijke conclusies. Ten eerste, deep learning is een heel nuttig voor het analyseren van microstructuren en vooral voor microstructuren die veel kleine details bevatten zoals de specifieke staalsoort van de tweede quiz. Ten tweede, het is perfect mogelijk om een deep learning model te trainen met slechts een beperkte hoeveelheid afbeeldingen en het is niet per se nodig om over een grote dataset te beschikken.

Eenmaal we een goede voorstelling van de microstructuren verkregen hebben met behulp van deep learning, kunnen we proberen om informatie over de behandeling, de microstructuur en de eigenschappen van het materiaal aan elkaar te linken. Concreet bestuderen we in welke mate het mogelijk is om op basis van één enkele microscopieafbeelding voorspellingen te doen over zowel de hardheid van het materiaal als de samenstelling. We trainen een deep learning model op een dataset van gelijkaardige staalsoorten. Het verkregen model gebruiken we vervolgens om microstructurele voorstellingen te bepalen van een tweede dataset met materialen waarvan we metingen hebben over de samenstelling en de hardheid. Omdat we vaststellen dat deep learning modellen niet goed veralgemenen naar nieuwe datasets, beslissen we om intermediaire output van het model te gebruiken om de microstructuur voor te stellen. Op deze manier, kunnen we het koolstofgehalte nauwkeurig gaan bepalen. Voor de hardheid en het mangaangehalte zijn de voorspellingen minder accuraat, maar nog steeds goed genoeg om een eerste afschatting te maken. Voor het siliciumgehalte zijn de voorspellingen heel slecht, wat lijkt te betekenen dat het niet mogelijk is om het siliciumgehalte te bepalen aan de hand van microscopieafbeeldingen. Door de voorspellingen van verschillende afbeeldingen uit te middelen, verkrijgen we accurate informatie over zowel het koolstofgehalte, het mangaangehalte en de hardheid van het materiaal.

Een laatste toepassing van deep learning die we bestuderen, is het artificieel verbeteren van de resolutie van afbeeldingen. Dit wordt ook soms artificiële super-resolutie genoemd. Deze klasse van modellen neemt als input een afbeelding met lage resolutie en maakt hier een hoge resolutie afbeelding van. Dit kan nuttig zijn om de hoeveelheid training data voor deep learning modellen te vergroten. We bestuderen de zogenaamde Unet-architectuur om de afbeeldingen twee- of viermaal te vergroten. Wanneer we de kwaliteit van de super-resolutie afbeeldingen visueel vergelijking met die van de hoge resolutie afbeeldingen, dan vinden we dat vooral voor de viermaal vergroting het deep learning model veel scherpere afbeeldingen oplevert dan conventionele interpolatie methodes. Wanneer we echter onderzoeken in welke mate de door het model geproduceerde afbeeldingen gebruikt kunnen worden in combinatie met andere machine learning modellen, vinden we dat het model dat de afbeeldingen viermaal vergroot amper beter scoort dan de conventionele interpolatie methodes. Hieruit concluderen we dat hoewel de super-resolutie afbeeldingen er goed uit zien, ze geen betrouwbare weergave van de werkelijkheid zijn. Langs de andere kant, vinden we dat het Unet dat de afbeeldingen tweemaal vergroot wel een duidelijk voordeel geeft ten opzichte van de meer conventionele interpolatie methodes. Vermits een tweevoudige toename van de vergroting leidt tot een viervoudige afname van

de tijd die nodig om de afbeeldingen te nemen, kan artificiële super-resolutie leiden tot serieuze tijds- en kostbesparingen.

We hopen dat dit werk de lezer overtuigt van de ongeziene mogelijkheden die machine learning heeft in de metallurgie en dat het de aanzet geeft tot het verder verkennen van de mogelijkheden van deep learning binnen dit onderzoeksveld.

Summary

The development of new materials has always been a key driver for technological advancement. In the Bronze age, civilizations that could produce bronze held a clear technological advantage over other civilizations. Later, in the Iron age, history repeated itself. Until today, iron plays an important role in our society. Especially steel, which is iron with less than 2.11 weight per cent carbon, is used in numerous markets such as the construction, mechanical equipment and automotive industry. More than 1 800 million tonnes of steel are produced on a yearly basis worldwide. Because of the widespread use, small improvements in the properties of steel can result in enormous savings. In order to systematically improve the properties of steel, it is necessary to fully understand the relation between the processing, the structure and the properties of the material. The structure of the steel at small scale, which is called the microstructure, is typically studied through microscopy imaging. However, it is not easy to link what one sees on such a microscopy image to the processing or the properties of the material. For decades, metallurgists have researched methods to extract a limited set of numbers from the microscopy image to characterize the microstructure more quantitatively.

In recent years, deep learning methods have become an invaluable tool for the analysis of images in fields such as medicine, biology and astronomy. One of the reasons why deep learning models are so good at analysing images is because they can learn which features in the image are important purely based on the data. This is different from earlier machine learning methods in computer vision, that required a manual extraction of features based on domain knowledge. A downside of deep learning is that it requires large datasets in order to be viable.

The aim of this PhD is to investigate which role machine learning and especially deep learning can play in the analysis of microscopy images. We start by investigating how deep learning can be used to represent the microstructure in a limited set of numbers. More specifically, we study a method called

triplet networks. This method learns to represent microstructure images in such a way that the representations of similar looking materials will lie close to each other, whereas visually very different materials will have representation that lie far away from each other. In other words, the distance in the representation space is a similarity measure. When we apply this method to a large dataset of sixty different materials, we find that by only extracting two number from each image the representations already contain enough information to differentiate between similarly looking materials. By extracting more numbers, it becomes even easier to differentiate between those materials. Thus, we conclude that triplet networks are a powerful method that can help metallurgists to describe microstructure images quantitatively. A disadvantage of the method, is that it mainly works for materials on which the triplet network was trained. For new materials, the amount of information contained in the representations is far less. However, we conjecture that this can be mitigated if a larger dataset containing more diverse microstructures is used.

The observation that deep learning models can easily differentiate between very similar looking microstructures that even expert metallurgists would struggle to tell apart, triggers an interesting question: is deep learning better at recognising microstructures than experts? To investigate this, two quizzes are organised in which both the experts and the deep learning model have to assign a number of question image to a set of predefined material classes. The first quiz consists of visually clearly different material classes, whereas the second focusses on a specific type of microstructures belonging to complex martensitic steels. In both quizzes, the deep learning model obtains a higher score than the average expert and in the second quiz the machine learning model even obtains a perfect score. This is remarkable as the deep learning model has only seen a dozen of example images for each material class during the training phase. Two things can be concluded from our results. One, deep learning can be very useful in the analysis of microstructures and especially for microstructures that contain a lot of small details such as the complex martensitic ones. Two, it is possible to train deep learning models in a very data-efficient manner so that we do not necessarily need very big datasets.

Once we have obtained good microstructural representation through triplet networks, we can try to establish a link between the processing, the structure and the properties. Concretely, we study to which extent it is possible to predict the composition and hardness of complex martensitic steels based on a single scanning electron microscopy image. We train a triplet network on a dataset of similar steels. The obtained network is then used to compute the microstructural representations of a second dataset of materials of which we have information about the composition and the hardness. As it turns out

that the triplet representations do not generalize well to new materials, we find that it is better to use intermediate output of the network as representation. Using this approach, we can accurately estimate the carbon content of the material. For the manganese content and the hardness, the predictions are less accurate, but still good enough to provide us with a meaningful first estimate. For the silicon content on the other hand, the predictions are not meaningful. This might indicate that it is not possible to determine the silicon content from microscopy images. By averaging the predictions of several images, we can obtain accurate information about either the carbon content, the manganese content and the hardness of the material.

A last application for deep learning we study is artificial resolution enhancement, also called artificial super-resolution. This class of models takes a low-resolution image as input and returns a high-resolution version of it. Artificially enhancing the resolution of images can be useful to increase the amount of training data for a deep learning model. We study the Unet architecture to achieve either twofold or fourfold magnification. Upon visual inspection of the super-resolution images, we find that especially the fourfold magnification model generates much sharper and more detailed images than conventional interpolation schemes do. However, when we investigate to which extent super-resolution images can be used in conjugation with other machine learning models, we find that the resolution enhancement of the fourfold magnification model is unreliable and does not offer an advantage over the original low-resolution images or other interpolation schemes. On the other hand, the twofold magnification model does add useful information to low-resolution images and offers significant advantages over other interpolation schemes. As a twofold increase in magnification results in a fourfold reduction in characterization time, artificial super-resolution can lead to significant time and cost savings.

We hope that this work convinces the reader of the incredible potential machine learning has in metallurgy and gives gusto to further explore the possibilities of deep learning in this research field.

Part I

Machine Learning for Microstructure Analysis of Steel

1

Introduction

Harder, Better, Faster, Stronger

Daft Punk

The aim of this PhD is to explore new possibilities by combining metallurgy, one of the oldest research fields in the world, with machine learning, a new and dynamic research field. Before discussing the aim of the project in more detail, we will first briefly introduce both fields.

1.1 Metallurgy and the steel industry

Saying that metallurgy is one of the oldest research fields is certainly not an understatement. Evidence has been found at archaeological sites mainly in present-day Serbia that already in 5 500 BC people were extracting metals from ores by heating them in fire.[1] Later, at around 3 500 BC, the first metal alloy was made in the Near East.[2] People discovered that by combining copper and tin, one could obtain bronze, which has favourable properties. This discovery is deemed so important by historians that it marked the beginning of a new era: the Bronze Age. It was only much later, at around 1 500 BC, that the Hittites developed a procedure to extract workable metals from iron ores.[3] They kept this procedure secret from the rest of the world until the demise of their kingdom at around 1 200 BC, which caused iron to become widely used in civilizations around the globe. The collapse of the Hittites marked the beginning of the Iron Age.

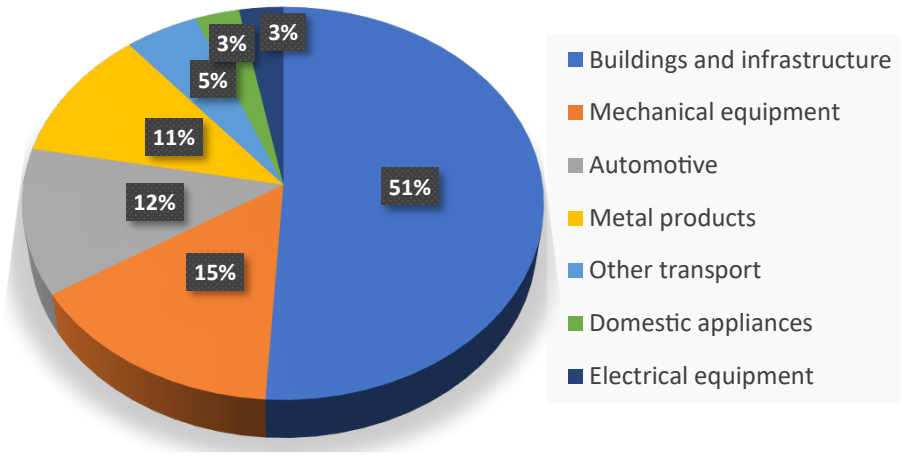


Figure 1.1: The different uses of steel in 2018. Data obtained from [4].

While the Iron Age contains the word iron, it is technically speaking more correct to call it the steel age. Steel is defined as iron with up to around 2 % of its weight (wt. %) consisting of carbon. The steel obtained through the procedure of the Hittites contained a small amount of carbon and would therefore be called steel nowadays. However, throughout the Iron Age none of the civilizations managed to accurately control the amount of carbon in the steel, so that high-quality steel remained scarce. The carbon content was either too low, resulting in a soft and bendable material or too high, so that the material was too hard and brittle for many applications. It was only two centuries ago that procedures were invented to accurately control the carbon content in steel. An example of such a procedure is the Bessemer process[5], in which impurities are removed from molten iron by blowing oxygen through it. It was also discovered that by adding additional alloying elements such as chromium, silicon or manganese the mechanical properties can be further improved. Additionally, the properties of steel can also be modified through different thermal treatments, such as rapid cooling (quenching) and controlled heating (tempering), or through mechanical deformations such as strain hardening.[6] Because of the wide range of mechanical properties that can be achieved, steel is nowadays used in numerous applications. Figure 1.1 shows the most important markets. We see that steel is mostly used in the construction of buildings and infrastructure. Steel is for instance commonly used in the construction of bridges, for the structural frame of skyscrapers and for rail tracks. Two other important markets are the mechanical equipment and automotive industry. Steel is among others essential for making bulldozers, tractors and cars.

In Figure 1.2, we show the evolution of the worldwide steel production since 2010. We see that the production of steel is still steadily increasing. In 2018, more than 1 800 million tonnes of steel were produced worldwide, a truly unfathomable amount. The European Union accounts for 9.3 % of the global steel production. The top producing countries are China (53.3 %), India (6.0 %) and Japan (5.3 %). The reason why steel is so commonly used, is because of its relatively low price. Carbon Fibre Reinforced Polymers (CFRPs) could for instance be a good alternative to steel in terms of weight and mechanical properties for many automotive applications. However, the price of such CFRPs is typically 10-12x higher than that of steel.[7] Because of this, CFRPs are currently only used in high-end applications such as Formula One racing cars.

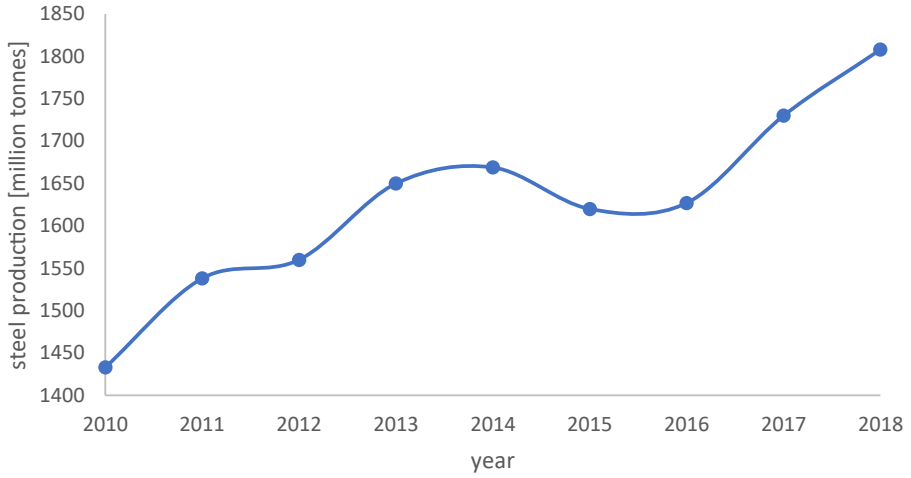
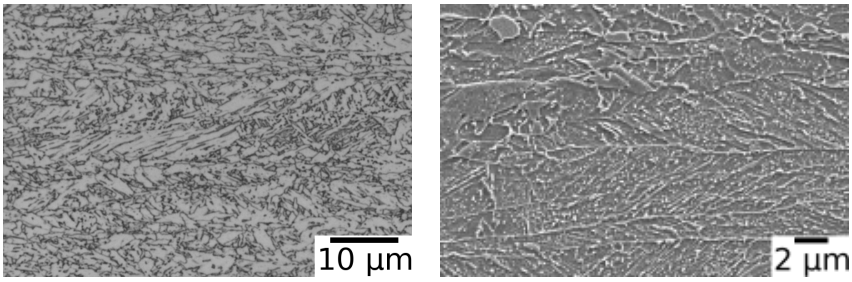


Figure 1.2: The evolution of the worldwide steel production in the past years. Data obtained from [4].

Because of the widespread use of steel, small improvements in the material properties can result in enormous savings. Historically, the properties of steel have been improved through trial-and-error. However, in order to improve the properties of a material efficiently, it is necessary to understand the relationship between the processing conditions and the properties. This relation is highly complex and most theoretical or empirical models fall short. By processing conditions, we refer in this work to both the composition and the thermal treatment of the steel. Linking the processing to the properties can be facilitated by considering that the properties are determined by the structure of the material and that the structure of the material is determined by the processing. In the literature, the strong interdependence between processing, structure and properties is referred to as the Processing-

Structure-Property (PSP) link.[8] Whereas the processing and the properties are measured values, the structure cannot simply be expressed in numbers. Typically the structure is analysed using microscopy imaging. In Figure 1.3, we show two examples of such microscopy images using different imaging techniques. The extraction of information from this kind of images is the central topic of this PhD. In the next section, we introduce machine learning to provide us with the necessary modelling tools to analyse microstructure images. More background to the metallurgical aspects of this work can be found in chapter 2.



(a) optical microscopy **(b)** scanning electron microscopy

Figure 1.3: Some examples of microscopy images of steel.

1.2 Machine learning

Wikipedia defines machine learning as "the study of computer algorithms that improve automatically through experience".[9] For this work, it suffices to think of experience as data. Machine learning methods allow us to model complex systems purely based on data. Unlike traditional science, where a profound understanding of a system is necessary in order to correctly model it, machine learning methods do not require any knowledge about the system one is modelling. This makes machine learning complementary to traditional scientific methods. Because of this, machine learning is sometimes called the fourth paradigm of science[10] which is illustrated in Figure 1.4. As is also shown in the figure, machine learning only gained major public interest in the late nineties, so it is still a very young field of research. Despite this, machine learning models are already commonly used in many fields including medicine[11], economics[12], biology[13] and physics[14].

Machine learning is typically subdivided into three categories as is shown in Figure 1.5. In supervised learning, a relationship between input and output variables is learned. The input variables are also called features, whereas the

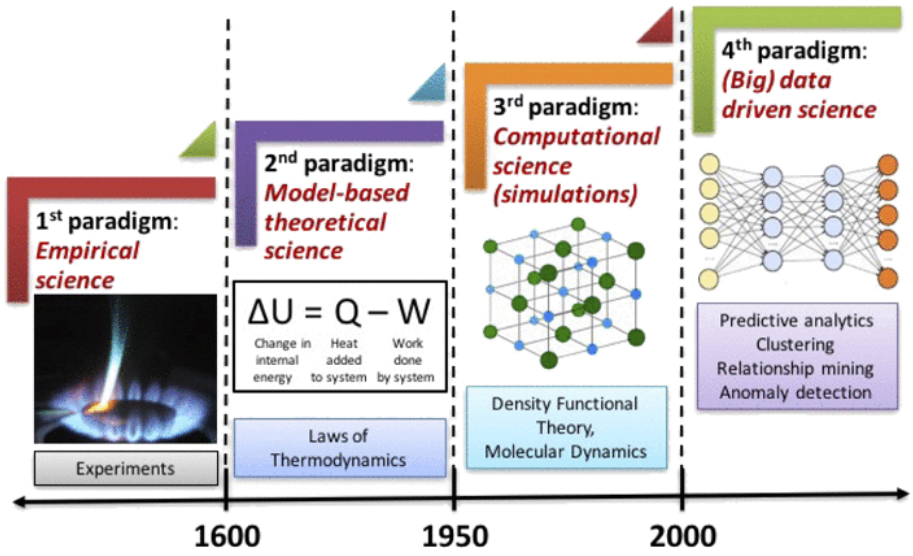


Figure 1.4: The four paradigms of science. Image taken from [10].

output variables are referred to as labels or targets. If the target variables are continuous, the problem is called a regression problem. If they are categorical, one considers a classification problem. A typical example of supervised machine learning is curve fitting. Supervised machine learning techniques are the most commonly used, as they are relatively straightforward to apply once a dataset has been collected. The problem with this set of techniques is that they can only be employed if all information about the input and output variables is available. In many practical situations this is not the case. Unsupervised machine learning algorithms focus instead on data without requiring information about the output variables. Typical examples include clustering, where similar data is grouped together, and dimensionality reduction, where redundant features are detected and removed. A last branch of machine learning techniques is called reinforcement learning. Here, the algorithm determines by itself which data is required and learns through trial and error. A popular example is the Alpha Go model developed by GoogleMinds.[15] The model was able to learn how to play Go at superhuman level solely by playing games against itself. While this example indicates the tremendous potential of reinforcement learning, the field has only recently gained traction and practical applications are still limited. We therefore do not consider any reinforcement learning methods in this work.

The main subject of this PhD is the analysis of microscopy images with machine learning. Such images typically have a pixel resolution of 1200 $\times$

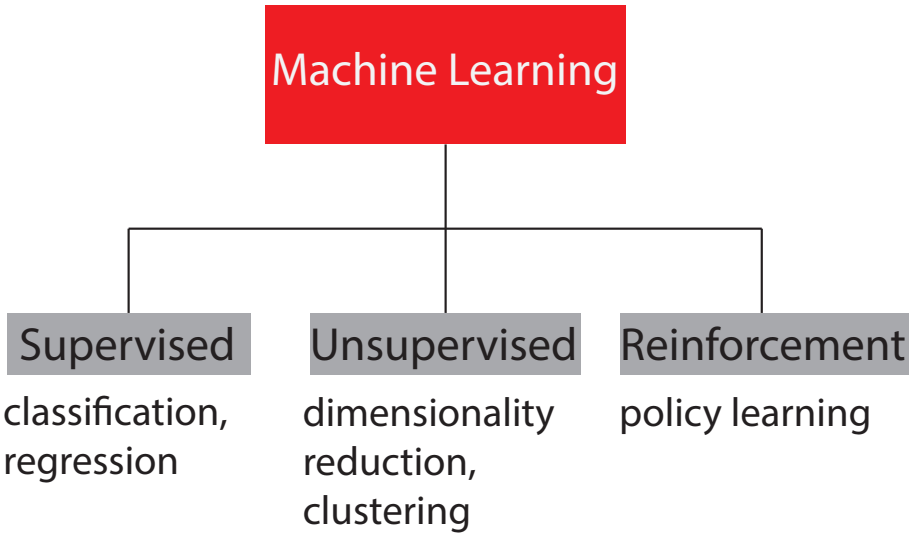


Figure 1.5: The typical subdivision of machine learning.

900 or more, so that an image can easily contain more than one million pixels. If one works with raw image data, the dimensionality of the input space can therefore be easily of the order of a million. Most machine learning methods are unable to cope with such high-dimensional inputs. Traditionally, this problem is solved by first extracting a set of features from the image. Engineering good features is an art on its own and requires a lot of domain knowledge. For the analysis of microscopy images of steel, a sensible approach is to use the same microstructural descriptors as metallurgists do. Information about the distribution of the grain sizes can for instance be extracted from the image and be used as input for a machine learning model. The downside of this approach, is that the relevant microstructural descriptors might differ strongly for different classes of steel, as it is for example not always possible to clearly discern or define grains in steel. One would therefore have to engineer different sets of features for different classes of steel, which is impractical. To deal with this problem, deep learning methods were developed.[16] This class of methods is able to deal with high-dimensional data by letting the machine learning model learn to extract relevant features from images by itself, thus overcoming the need for feature engineering. Deep learning is nowadays one of the most important subfields of machine learning.

The terms machine learning, deep learning and artificial intelligence are often used interchangeably. In many cases, this is correct, although artificial intelligence is strictly speaking a more general term, as machine learning is

considered to be a subfield of AI. Although AI has many definitions, most textbooks define AI as “the field as the study of rational agents: any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals”.[17] In order to maximize the chance of success, AI relies on machine learning methods to improve from experience. AI also makes use of for instance logic, graph theory and mathematical optimization in order to maximize the chance of success. In Figure 1.6, we show the number of search queries on google for the terms "machine learning", "deep learning" and "artificial intelligence". It is clear that in the past decade the interest in machine learning has exploded. Both industry and academia have been attracted to the potential of machine learning to extract new patterns and insights from available datasets. Machine learning and artificial intelligence have been commonly used as buzz words and can currently be considered as a real hype in many fields. However, we see that in the past year the number of search queries has been slightly decreasing, which might indicate that the hype is fading. It will be interesting to see to which extent the surge of machine learning will have long-lasting impact. We hope that at least the presented research will prove its relevance now and in the future. More details on the machine learning methods will be provided in chapter 3 and in the relevant result chapters.

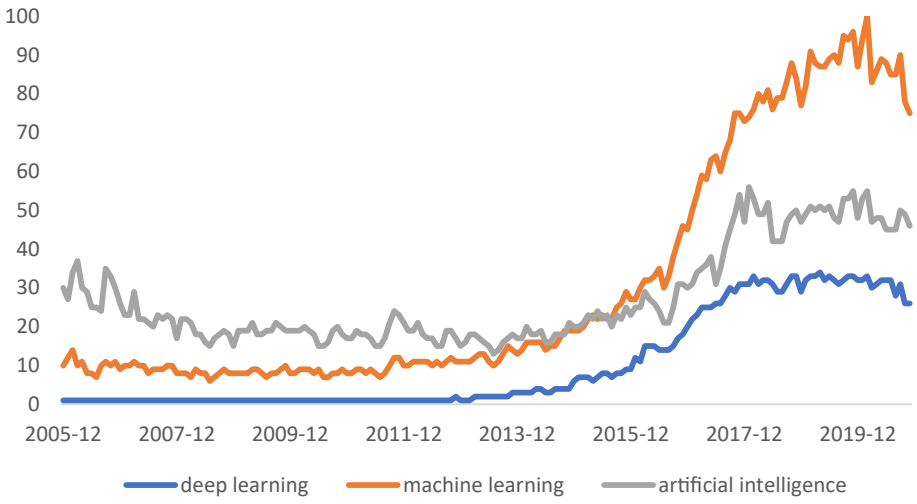


Figure 1.6: The evolution of the number of search queries on Google for the terms "machine learning" and artificial intelligence" normalised to the maximum number of search queries for "machine learning". Source: Google Trends.

1.3 Project aim and overview of this work

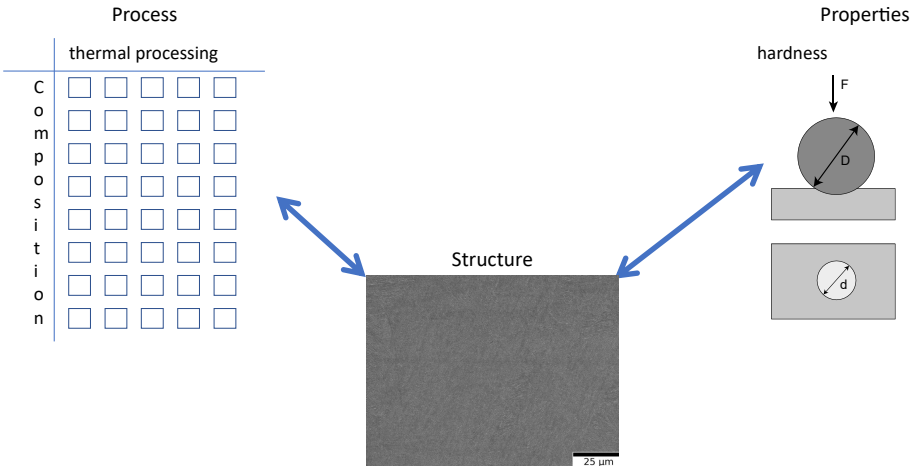


Figure 1.7: The aim of this project is to investigate how machine learning can help to model the links between the processing, the structure and the properties of steels.

The aim of this project is to study which role machine learning can play in the development of new steels and especially in better understanding the links between the processing, the structure and the properties. Because of the great success of deep learning in many computer vision tasks, we mainly focus on the analysis of microscopy images using deep learning. The first question we ask ourselves is how deep learning can be used to represent the microstructure in a limited set of numbers. This is highly relevant question, as for machine learning models it is much easier to deal with a limited set of numbers than with raw image data. Different microstructural representations are studied in chapter 4. Once we have obtained good, low-dimensional microstructural representations, it is interesting to study how much detailed information they contain. To test this, we represent the microstructure with only two numbers and train a machine learning algorithm on these numbers. We show that by only using these two numbers the machine learning model can already recognise different kinds of microstructures as well as the best human experts. These results are reported in chapter 5. Armed with these powerful representations of the microstructure, we then investigate to which extent it is possible to establish links between the processing, the structure and the properties, which is schematically illustrated in Figure 1.7 For the processing, we mainly look at the composition, whereas for the properties we are primarily interested in the hardness. We consider a specific set of complex martensitic steels, for which the composition and thermal treatment

is accurately recorded. For all materials, different microscopy images are taken and the hardness is measured. We investigate how accurately we can predict the composition and the hardness of a material based on a single microscopy image. This is discussed in chapter 6. A last topic we study, is that of artificial resolution enhancement. Here, we train a deep neural network to output a high-resolution version of a low-resolution input image. The network thus learns to artificially increase the magnification of microscopy images. This can potentially lead to faster material characterization and can augment the amount of image data that is available to train deep learning models on. The findings of this study are reported in chapter 7.

2

A quick guide to metallurgy

There are three things extremely hard: steel, a diamond, and to know one's self.

Benjamin Franklin

Although the emphasis of this work lies on the machine learning methodology, a basic knowledge of materials science and metallurgy is required to understand the research results. In this chapter, we therefore aim to provide the reader with a succinct background in metallurgy. We first discuss material properties in general, after which we focus on the processing and characterization of steel.

2.1 Material Properties

Before introducing the relevant metallurgical aspects, we discuss the most important material properties for this work. We focus on the operational definitions which are necessary to understand the results in later chapters. A more detailed description of these properties requires the introduction of the proper constitutive equations which is beyond the scope of this chapter. We refer the interested reader to introductory textbooks on continuum mechanics.[18]

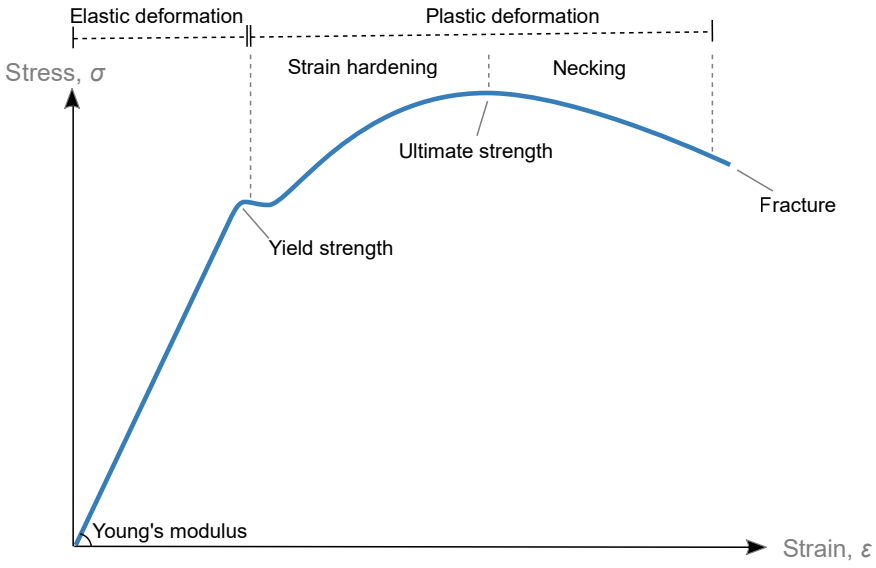


Figure 2.1: The stress-strain curve for a low carbon steel. Image adapted from [19].

2.1.1 Stress-strain curve derived properties

The stress-strain curve of a material is obtained by gradually applying a load to a material and measuring its deformation.[20] This procedure reveals many relevant material properties, which we will briefly discuss here. Figure 2.1 shows a typical example of a stress-strain curve for a low carbon steel bar under tension. If we denote F the applied load, A_0 the original cross-section of the bar, L_0 the initial length and L the length after the application of the load, the engineering stress and strain are defined by:

$$\sigma = \frac{F}{A_0}$$
$$\varepsilon = \frac{L - L_0}{L_0} = \frac{\Delta L}{L_0}$$

Under sufficiently small stresses, the bar will deform reversibly. This type of deformation is called elastic deformation and the relation between the stress and the strain is approximately linear:

$$\sigma = E\varepsilon,$$

where the slope E of this relation is Young's modulus. The maximal stress that can be applied under elastic deformation is the **yield strength**. For higher stresses, the deformation becomes irreversible and is called plastic

deformation. These deformations cause the motion and generation of dislocations in the crystal lattice which allow the material to resist higher stresses. This process is called strain hardening and is a common procedure to increase the yield strength of a material. The highest stress that can be applied in this regime is called the **ultimate strength**. For even higher stresses, the cross-section of the material starts to reduce at certain regions of the material, which is called necking. The neck refers to the region of the material with the reduced cross-section. By further increasing the stress, the cross-section of the neck decreases until a fracture of the material occurs.

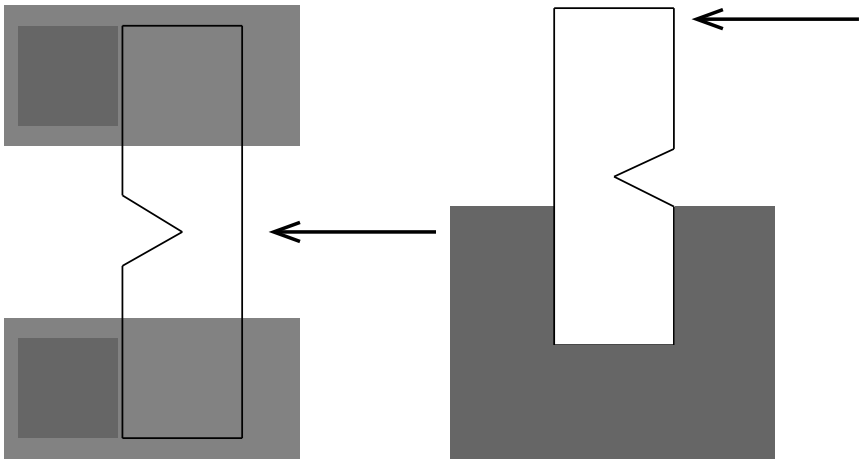
Two other relevant material properties can be derived from the stress-strain curve. **Ductility** is defined as the amount of deformation a material can bear before fracture occurs. It is commonly quantified as the percent elongation (%EL):

$$\%EL = \frac{L_f - L_0}{L_0},$$

where L_0 is the initial length and L_f is the length at the moment the fracture takes place. A material is typically considered to be ductile when it has a %EL of above 5%[21], while for lower values the material is said to be **brittle**. **Toughness** is the amount of energy that a material can absorb. It is given by the area under the stress-strain curve. Materials with high ductility and strength will therefore also be tough. Toughness is typically measured using an impact test, where a pivoting arm is dropped from a specific height. The arm swings down and breaks the sample, which is notched to assure the break occurs at a fixed position. By measuring the height after the swing, the absorbed energy and hence the toughness can be computed. Two commonly used measuring methods are the Charpy[22] and Izod[23] impact tests, which are shown in Figure 2.2.

2.1.2 Hardness

The most important property for this work is hardness. There are three main types of hardness[24]: scratch hardness, rebound hardness and indentation hardness. All three types have in common that they express the resistance of the material to plastic deformation. Scratch hardness expresses how easy it is to scratch a material. In mineralogy, the well-known Mohs scale is used to measure this. The scale relies on the fact that a material is harder than another material if the former is able to make a scratch on the latter. In metallurgy, it is usually measured using a sclerometer. This device uses a diamond under a fixed load to make a scratch on the sample material. The width of the scratch is then a measure for the hardness of the material. Rebound hardness, also known as dynamic hardness, measures the rebound



(a) Charpy **(b)** Izod

Figure 2.2: Two common methods to measure the toughness. For the Charpy test, we show the top view, whereas for the Izod test the frontal view is shown.

height when a diamond tip is dropped from a fixed height onto the sample material. The last type of hardness is the indentation hardness, which measures how easily a material can be indented. This type of hardness is the focus of this work.

Indentation hardness is commonly measured by indenting the material with an indenter under a fixed load. Two commonly used scales for hardness are the Vickers[25] and Brinell[26] hardness, for which the testing conditions are shown in Figure 2.3. The pyramid that is used for the Vickers hardness test is typically made of diamond, whereas the ball that is used for the Brinell hardness testing is usually made out of steel or tungsten carbide.

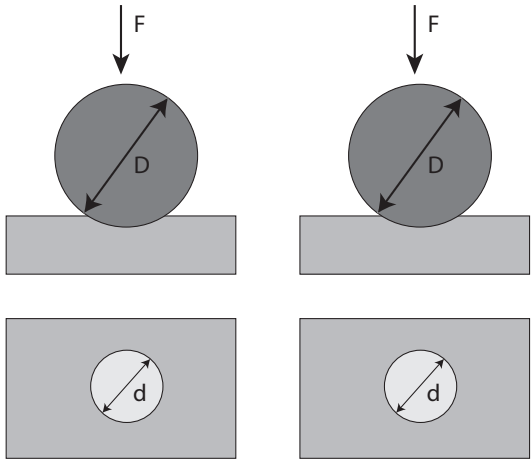
The Vickers hardness (HV) is computed as:

$$HV = \frac{F}{A},$$

where the surface area A is obtained as:

$$A = \frac{d^2}{2\sin(136°/2)},$$

with F the applied load expressed in Newtons and d the diagonal of the indentation expressed in millimetres.



(a) Vickers **(b)** Brinell

Figure 2.3: Two common methods to measure the indentation hardness. Images adapted from [27] and [28] respectively.

The Brinell hardness (HB) is computed as:

$$\mathrm{HB} = 0.102 \frac{2F}{\pi D \left(D - \sqrt{D^2 - d^2} \right)}$$

with F the applied load expressed in Newtons, D the diagonal of the ball expressed in millimetres and d the diameter of the indentation expressed in millimetres.

Since in both cases the hardness is given by the ratio of a force to an area, hardness can be expressed in pascals. However, it is not a pressure, as for hardness one considers the surface area rather than the area orthogonal to the direction of the force.

Different values can be taken for the load. When a load is equal or higher than 9.81 N, one considers a macroindentation test. For loads lower than this value, the tests are called microindentation tests. In the latter case, the measured hardness is often referred to as "microhardness".[24]

Figure 2.4 shows the relation between the Brinell and Vickers hardness for steels. For the range of 100-500 HB there is an almost linear relation between both. For higher hardnesses, this is no longer the case. This observation is important, as it shows that it is not ideal to combine measurements in Brinell and Vickers hardness into one dataset to train machine learning models.

As we mentioned before, hardness expresses the level of resistance of a material against plastic deformations. This is also the case for the tensile

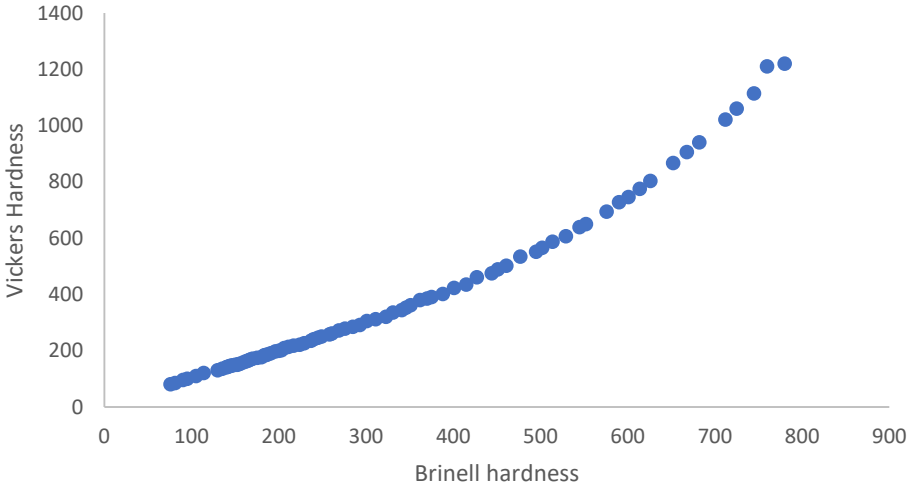
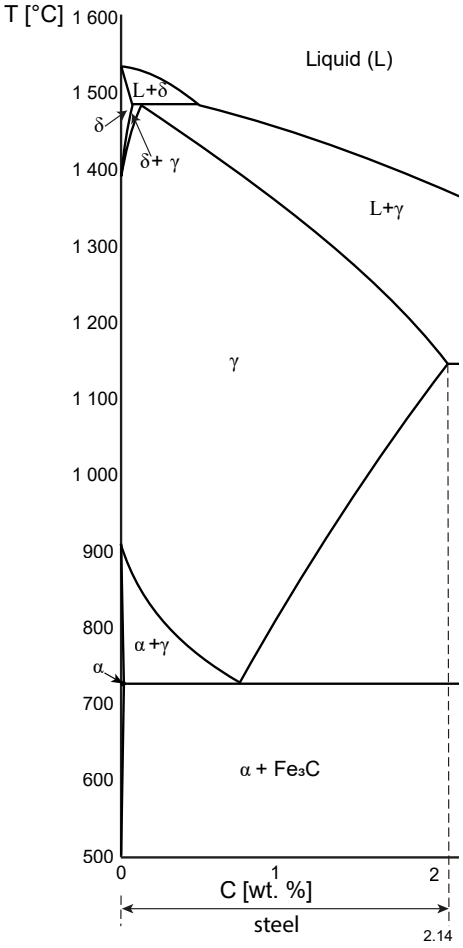


Figure 2.4: The relation between Brinell and Vickers hardness for steels. The Brinell hardness was measured using a 10 mm ball and a 29,430 N load. Data obtained from [29].

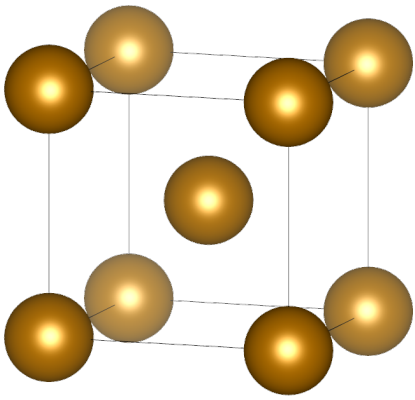
strength. Therefore, one would expect that both properties are positively correlated with each other. Pavlina *et al.*[30] empirically show that this is the case and suggest that an approximately linear relationship between both properties exists.

2.2 Iron and steel

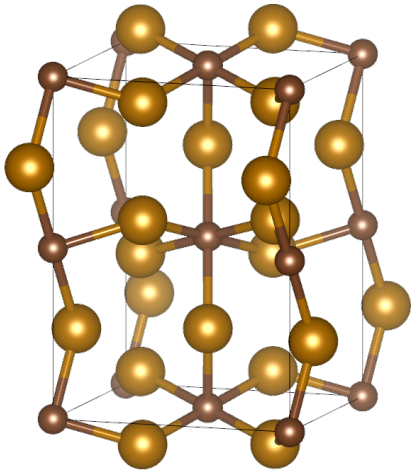
Iron is by mass the fourth most common element in the Earth's crust.[31] At room temperature and standard pressure, it has a body-centered cubic (bcc) crystal lattice and is called alpha iron or ferrite.[32] It is a relatively soft and ductile material that is ferromagnetic. In the presence of water and oxygen, alpha iron oxidizes and forms different types of iron oxides such as magnetite, wüstite and hematite. When hematite is mixed with metallic elements such as strontium or barium, the resulting material is also called ferrite. This material is also ferromagnetic and has applications as inductors, transformers and antennas. When alpha iron is heated above 912 °C, a phase transformation occurs and the lattice structure changes into a face-centered cubic (fcc) lattice. The resulting phase is called gamma iron or austenite. It has a similar hardness and ductility as ferrite. At 1 394 °C, a new transformation occurs and austenite is transformed into delta iron, which has again a bcc lattice. The melting point of iron is at 1 538 °C.



(a) The iron-carbon phase diagram for steel. Image adapted from [35].



(b) α-ferrite



(c) cementite (F_3C)

Figure 2.5: The iron-carbon phase diagram for steel and the crystal lattices of the two thermodynamically stable steel phases at room temperature.

By adding a small amount of carbon to iron, one obtains steel. The iron-carbon phase diagram is shown in Figure 2.5a for the relevant range of carbon.[33] The addition of carbon strongly affects the temperature at which austenite is formed. The formation temperature of austenite reaches a minimal value of 723 °C at around 0.8 wt. % C. The corresponding point on the diagram is called the eutectoid point. Below the eutectoid temperature, two main phases of steel occur. For less than 0.02 wt.% carbon, one has pure ferrite of which the crystal lattice is shown in Figure 2.5b. As more carbon

is absorbed, cementite or iron carbide is formed. In its pure form this a ceramic material with an orthorhombic crystal lattice that contains 6.67 wt. % carbon. The crystal lattice of cementite is shown in Figure 2.5c. Cementite is hard and brittle. It is typically obtained by slowly cooling austenite. As more carbon can be dissolved in austenite than in ferrite because of its fcc lattice, carbides are formed due to the excess of carbon in ferrite. Carbides can occur in the form of small grains, but they can also appear in lamellar structures. In that case, the structure is called pearlite, which is a mixture of 88.5 wt. % ferrite and 11.5 wt. % cementite arranged in alternating layers of both phases. It is well-known for its exceptional tensile strength, which can be above 6 GPa, making it one of the strongest structural bulk materials on earth.[34] Because of this, pearlite is often used in wires for instance for the suspension of bridges.

The phase diagram shown in Figure 2.5a might give the impression that austenite is irrelevant in most practical applications, as it only occurs above 723 °C. This is however not the case. The phase diagram only shows which phases of steel are thermodynamically stable at a given temperature and carbon content. Information about the kinetics is not included in such a figure. In the following section, we will discuss the kinetics of steel in more detail.

2.3 Away from the equilibrium

When austenite is cooled rapidly, neither the iron atoms nor the carbon atoms have time to diffuse. This is shown in Figure 2.6 by the red curve labelled "Fast Cool". Due to the cooling, the fcc lattice of austenite transforms into a body-centred tetragonal lattice. This lattice has less space to accommodate the carbon atoms and is therefore supersaturated with carbon which puts a strain on the lattice. Furthermore, the transformation induces shear deformations that cause many lattice defects and dislocations. The resulting phase of iron is called martensite. Due to the presence of the dislocations and lattice strains, it is considerably harder than pearlite achieving a hardness of up to 700 Brinell, while it is much more brittle.[36] Depending on the composition and the thermal processing not necessarily all austenite undergoes a phase transformation. The part of the austenite that does not transform is called retained austenite.

While the hardness of martensite is often desirable, its lack of ductility and toughness can be troublesome for many applications. This can be partially solved by tempering the martensite, i.e. by reheating it to 300-600 °C. The higher temperature allows the carbon atoms to escape from

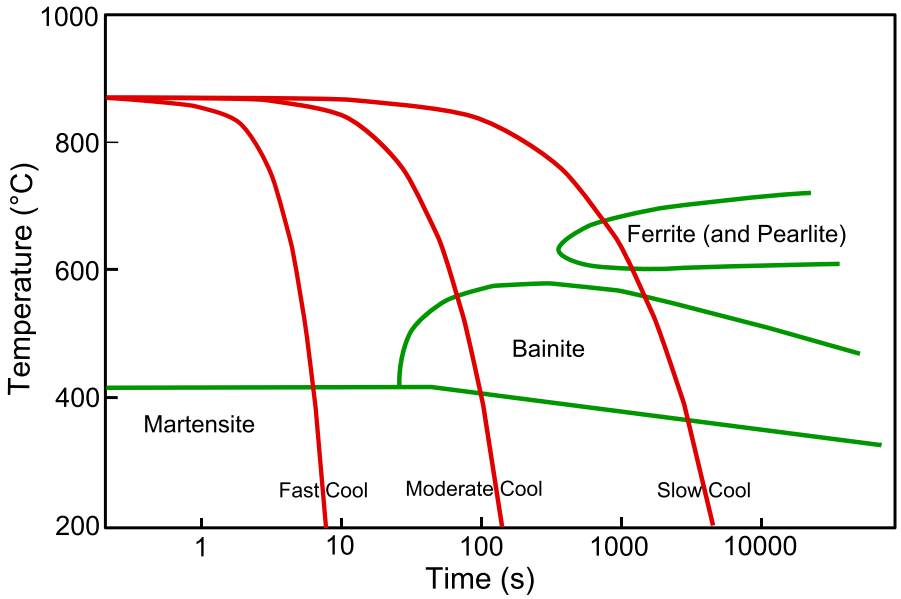


Figure 2.6: A typical continuous cooling transformation (CCT) diagram for a steel alloy. Image adapted from [37].

the supersaturated martensite, thus relieving the strain on the lattice. By reducing the strain on the lattice, it becomes easier to plastically deform the tempered martensite causing the ductility and toughness to increase. One would expect this to come at the expense of a serious reduction in hardness, but this is not always the case. The carbon that has migrated from the martensite might form small carbide precipitates. These precipitates resist plastic deformations in the material, causing the hardness to remain high. This type of heat treatment is called precipitation hardening and is a good example of how heat treatments such as tempering can be applied to obtain steel with a more desirable set of properties.

When cooling at moderate cooling rates, as shown in Figure 2.6, bainite is formed.[32] The iron atoms still do not have time to diffuse, but the carbon atoms do. They migrate from the strained bcc lattices towards regions with retained austenite or precipitate to form cementite. This is shown schematically in Figure 2.7, where a further distinction is made between lower and upper bainite. For upper bainite, which is formed at temperatures ranging from 400 °C to 550 °C, the formation originates from ferrite nuclei. Precipitation from the carbon-rich austenite layer helps these nuclei to grow. The resulting carbides tend to lie parallel to the long axis of the ferrite grain which results in a feathery structure. Lower bainite is formed at temperatures

between 250 °C and 400 °C. After the formation of supersaturated ferrite, carbides precipitate within the needle-shaped ferrite grain in a plate-like form. The orientation of these platelets alternates between the different ferrite grains with an angle of 55 °. In the literature, it is found that ferrite-bainite dual phase steels tend to have lower strength and hardness, but higher ductility compared to ferrite-martensite dual phase steels.[38]

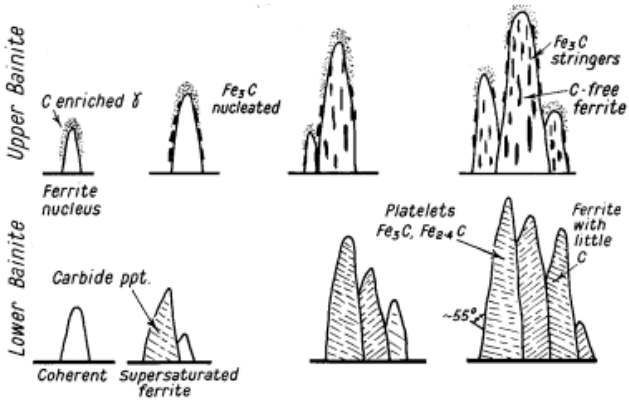


Figure 2.7: A schematic illustration of the formation of bainite. Image adapted from [39].

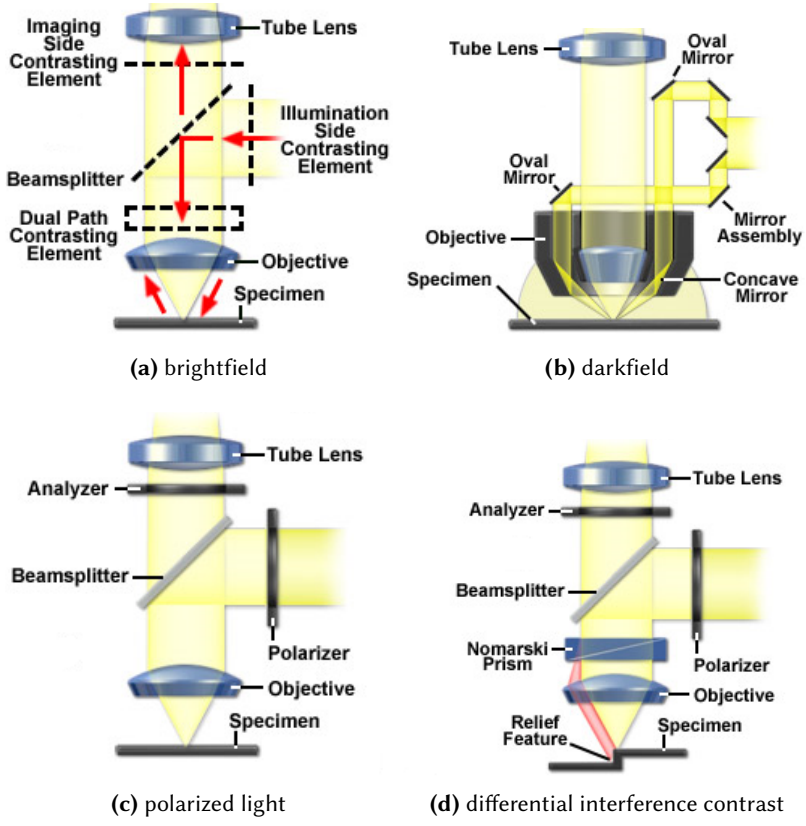
So far we have only discussed carbon as an alloying element in steel. In practice, many different alloying elements are used, each with their own effects on the properties and phase transformations. Elements such as manganese or nickel have a stabilising effect on the austenite content and when they are added in sufficient concentrations they can completely replace the ferrite phase by austenite at room temperature.[32] Hence, these elements are said to stabilise austenite. At lower concentrations, they can also shift the CCT diagram to the right, allowing martensite to be formed at much slower cooling rates. Elements such as silicon and aluminium on the other hand promotes the formation of ferrite. The presence of these elements can strongly reduce or even completely eliminate the retained austenite in the steel. Furthermore, at concentrations of above 2 wt. % Si, the mixture of iron-graphite is thermodynamically preferred over ferrite-cementite, resulting in a material that has excellent wear properties.[6]

It should be clear that steel can occur in many forms. Each form has its own distinct advantages in terms of properties. By cleverly conceiving thermal treatments and by adding various alloying elements, metallurgists can combine different forms of steel in a single material to obtain the right set of properties for specific applications. In order to do so successfully, it is necessary to analyse the structure of the steel at a small scale, the

microstructure. In the next section, we briefly discuss the primary tool to do so: microscopy imaging.

2.4 Microscopy imaging of steels

Two types of microscopy imaging are relevant for this work: optical microscopy (OM) imaging and scanning electron microscopy (SEM) imaging.



(a) brightfield **(b)** darkfield

(c) polarized light **(d)** differential interference contrast

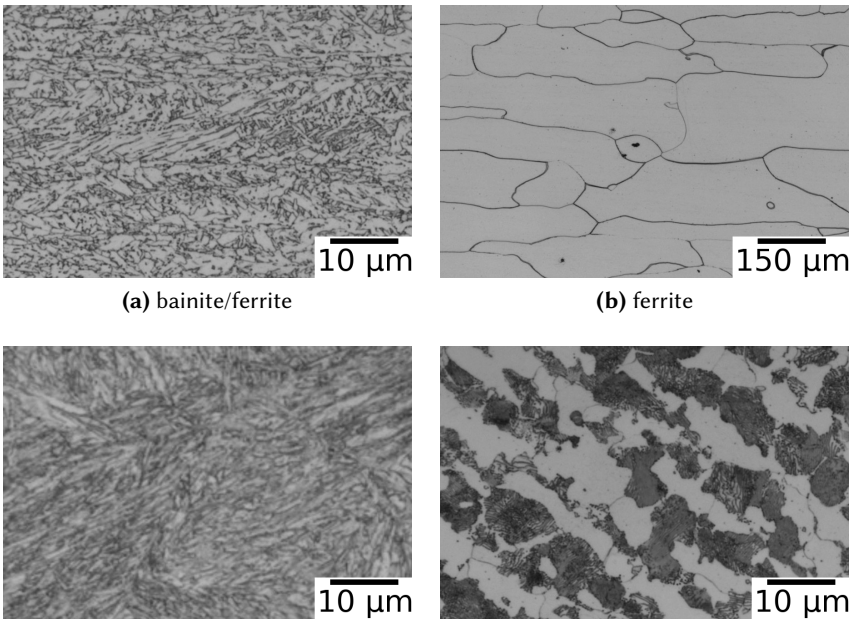
Figure 2.8: An illustration of some commonly used optical microscopy techniques used in metallurgy. Image adapted from [40].

In Figure 2.8, we show some of the most commonly used optical microscopy imaging techniques in metallurgy. The simplest setup is the brightfield configuration (a). A condensed light bundle is reflected on the sample and is detected at the point denoted "Tube Lens" in the figure. Some examples of brightfield OM microstructure images are shown in Figure 2.9. Darkfield

microscopy (b) has a slightly more complicated configuration. By relying on a mirror array with oval opening and a concave mirror, the light reaches the sample at a high incidence angle. If there is no topography present in the sample, the light will not be reflected into the detector and the image remains dark. However, areas with relief features will cause light to be reflected into the detector leading to bright features. Darkfield microscopy is therefore ideally suited for examining the relief of a sample. The reflected polarized light configuration (c) is used for samples that alter the polarization upon reflection, as is the case in certain metallic alloys. As the name suggests, this method makes use of polarized light to examine the sample. The unpolarised light from the light source first passes through a polariser. After reflection on the sample, the light passes through a second polariser, called analyser in the figure, which is oriented at 90 ° with respect to the first polariser. Hence, only the light that has been depolarized due to reflection on the sample passes through the detector. A similar approach is taken in the Differential Interference Contrast (DIC) configuration (d). The main difference with the polarized light configuration is that the light beam has to pass through a Nomarski prism before going through the objective. This prism splits the light beam into two orthogonally polarized beams that laterally reflect on the sample. If the sample has no relief, the beams will cancel each other out and no signal is recorded. However, if there is even the slightest bit of relief in the surface, there will be a phase difference in both beams due to the different path lengths the beams have to travel. This phase difference can then be converted at the analyser into a grey value image.

From the discussion above, it should be clear that optical microscopy works best when there is some relief in the sample. To achieve this, the sample is usually etched with a chemical agent, which is called the etchant. The etchant helps to reveal the grain boundaries and the different steel phases in the sample. Many kinds of etchants are used in metallurgy. Nital etching, which consists of 1-10 % nitric acid and 90-99 % ethanol or methanol, is most commonly used for the etching of ferrite and steels. Other commonly used etchants are Picral, which is mainly used for samples containing ferrite and cementite phases, Villela, which is used for ferrite and carbide structures and Bechet and Beaujard, which is mainly used for martensite and bainite.[41]

The resolution that can be achieved with optical microscopy is limited due to the diffraction limit, which states that the maximal resolution of an imaging system is proportional to the wavelength of the detected light. Scanning electron microscopy makes use of electrons rather than light. As electrons have a wavelength that is about 1 000 times smaller than that of visible light, the diffraction limit is typically no longer the limiting factor for the resolution of SEM imaging.[43] The factors limiting the resolution in SEM are rather the



(a) bainite/ferrite **(b)** ferrite

(c) (tempered) martensite **(d)** pearlite/ferrite

Figure 2.9: Some examples of OM microstructure images for different structures of steel.

spot size of the electron beam on the sample and the volume of the sample that interacts with the electron beam, also called the interaction volume.[43] The typical configuration of a scanning electron microscope is shown in Figure 2.10. Electrons are thermionically emitted from a filament that serves as the cathode. The filament is commonly made of tungsten, as it has the highest melting point of all metals and because of its low thermal expansion coefficient which is necessary for a stable electron beam. Due to the presence of an electric field, the electrons accelerate towards the anode. A magnetic lens, which consists of a set of electromagnets, focusses the electrons. The electrons pass through the scanning coil that directs the electrons towards a specific point on the sample. This coil permits to scan points on the sample in a raster-like fashion, so that every scanned point corresponds to one pixel on the resulting image. The electrons collide with the sample and are either backscattered or collide with an electron in the sample causing the emission of secondary electrons and possibly X-rays. The backscattered electrons are detected and they reveal information about the composition of the sample. Heavy elements will backscatter the incoming electron more strongly and will therefore appear brighter in the image. The secondary electrons are mainly used to chart the relief of the sample, as regions with a high relief will emit

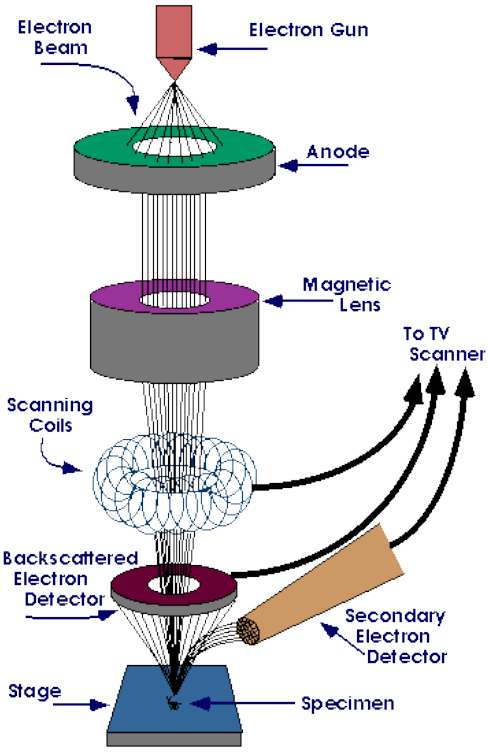
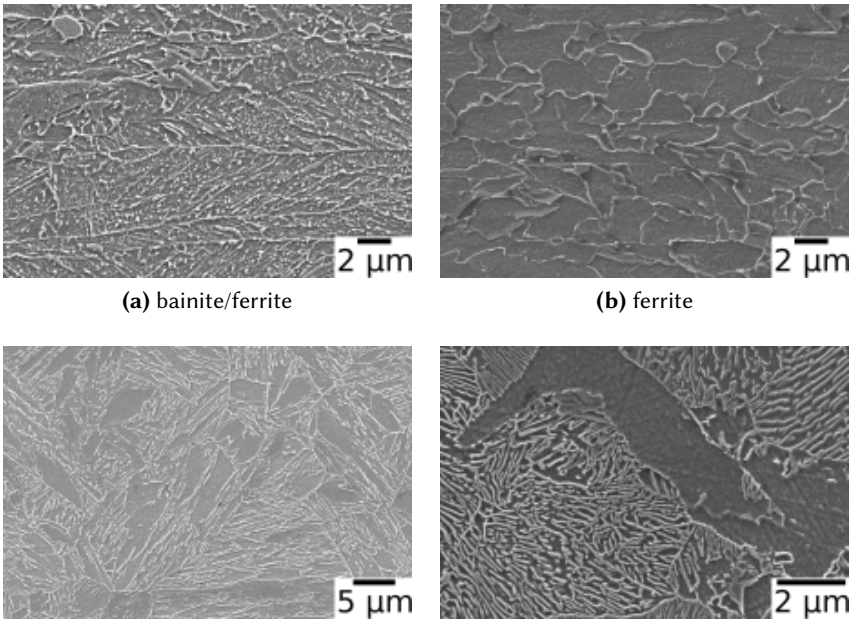


Figure 2.10: A schematic illustration of a scanning electron microscope. Image taken from [42].

more secondary electrons.

An important modification to the SEM imaging described above, is the use of a Field Emission Gun (FEG).[43] This type of electron gun has a sharp point with a radius of about 100 nm which allows the electrons to tunnel when an electric field with an extraction voltage of a few kV is applied. Unlike with thermionical emission, the emitted electrons are more coherent and all go in the same direction leading to an increase in current density of up to three orders of magnitude. This results in an improved signal-to-noise ratio and a better spatial resolution. In Figure 2.11 we show some examples of FEG SEM images.



(a) bainite/ferrite **(b)** ferrite

(c) (tempered) martensite **(d)** pearlite/ferrite

Figure 2.11: Some examples of SEM microstructure images for different structures of steel. The top row shows FEG SEM images, whereas the bottom row shows Tungsten SEM images.

3

Computer vision and deep learning

In this chapter we aim to provide the reader with a succinct overview of computer vision methods. We make the distinction between what we call classical computer vision, where features are extracted from the image according to a fixed recipe, and deep learning, where the model decides for itself which features to extract from the image. The emphasis is put on the deep learning part as it is the most important for this work.

3.1 Classical computer vision

We discuss two types of classical computer vision features: Haralick features and two-point statistic features. Both have been applied to the analysis of microstructure images in the literature.

3.1.1 Haralick features

Haralick features[44] are computed based on the Gray-Level Co-occurrence Matrix (GLCM). We denote by X a greyscale image with a pixel resolution of $H \times W$ and with each pixel having a N-bit representation, so that there are 2^N possible grey levels. For a fixed offset $\mathbf{d} = (d_x, d_y)$, the (non-normalised) GLCM is defined as:

$$P'(i,j) = \sum_{h=1}^{H} \sum_{w=1}^{W} \begin{cases} 1, & \text{if } X(h,w) = i \text{ and } X(h+d_x, w+d_y) = j, \\ 0, & \text{otherwise,} \end{cases}$$

descriptor	formula
energy	$\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)^2$
contrast	$\sum_{i=1}^{N}\sum_{j=1}^{N} (i-j)^2 P(i,j)$
correlation	$\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{(i-\mu_x)}{\sigma_x}\frac{(j-\mu_y)}{\sigma_y} P(i,j)$
variance	$\sum_{i=1}^{N}\sum_{j=1}^{N} (i-\mu)^2 P(i,j)$
homogeneity	$\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{P(i,j)}{1+(i-j)^2}$
sum average	$\sum_{k=2}^{2N} k P_{x+y}(k)$
sum entropy	$-\sum_{k=2}^{2N} P_{x+y}(k)\log P_{x+y}(k)$
entropy	$-\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)\log(P(i,j))$
difference average	$\sum_{k=0}^{N-1} k P_{x-y}(k)$
difference entropy	$-\sum_{k=0}^{N-1} P_{x-y}(k)\log P_{x-y}(k)$
information measure of correlation 1	$\frac{H_{xy}-H_{xy1}}{\max(H_x,H_y)}$
information measure of correlation 2	$(1-\exp(-2(H_{xy2}-H_{xy})))^{\frac{1}{2}}$
maximal correlation coefficient	$\sqrt{\lambda_2\left(\sum_k \frac{P(i,k)P(k,j)}{P_x(k)P_y(k)}\right)}$

Table 3.1: The fourteen Haralick descriptors and their definition. See the text for more explanation.

The GLCM counts the amount of times the grey values i and j occur simultaneously in pixels removed from each other at a pixel distance $\mathbf{d}$. Typically, the normalised GLCM is used, which is defined as:

$$P(i,j) = \frac{P'(i,j)}{\sum_{k=1}^{H}\sum_{l=1}^{W} P'(k,l)}.$$

The normalised GLCM can be interpreted as a probability distribution of grey value pairs.

This distribution can be characterized by a set of descriptors. Haralick proposed to use the 14 descriptors which are given in Table 3.1. In the table, μ is the mean of the joint distribution, whereas μ_x and μ_y are the means for the respective marginal distributions. This is analogous for σ, which denotes the standard deviation. The distributions P_{x+y} and P_{x-y} represent the probability of co-occurrence for entries in the GLCM of which respectively the sum and the difference of the coordinates is the same. H_{xy} denotes the entropy of the joint distribution, whereas H_x and H_y refer to the entropy of the corresponding marginal distributions. H_{xy1} and H_{xy2} are

defined as

$$H_{xy1} = -\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)\log(P_x(i)P_y(j))$$
$$H_{xy2} = -\sum_{i=1}^{N}\sum_{j=1}^{N} P_x(i)P_y(j)\log(P_x(i)P_y(j)).$$

Lastly, $\lambda_2(\mathbf{Q})$ denotes the second highest eigenvalue of the matrix $\mathbf{Q}$. The last descriptor is usually omitted due to numerical instability. The Haralick features as they are presented here are not rotationally invariant due to the fixed offset in a certain direction. To mitigate this, the descriptors are typically computed in four different directions and averaged. In the literature Haralick features have been successfully applied to the problem of microstructure recognition in a number of cases.[45, 46] Webel *et al.*[46] use a variation of the Haralick features, called textural features, in which they only consider the energy, contrast, correlation and homogeneity. They extract these descriptors from rotated versions of the image and compute for each descriptor the amplitude and the mean over the different rotations. Besides Haralick features, local binary patterns (LBP) have also been used in the literature to represent microstructure images.[45] This method also considers the immediate neighbourhood of the pixels in the greyscale image and keeps track of how many times each type of neighbourhood occurs. Another method that has been used is the histogram of oriented gradients (HoG).[45] This method considers the magnitude and direction of the gradients of every pixel. For both quantities a histogram is made for the entire image. Finally, both histograms are concatenated to form a single feature vector for the image. Threshold adjacency statistics (TAS) are a relatively recent method that has been used to extract features from microstructure images. These features are obtained by thresholding the image and counting the different neighbourhoods of the white pixels in the image. The neighbourhood is characterized through the number of white pixels it contains. If one considers the eight closest neighbours of a pixel, the number of different neighbourhoods is nine since the number of adjacent pixels that are white ranges from zero to eight. This method thus results in nine features that characterize the image.

Another important class of visual features that are used to describe microstructural images is the visual bag of words.[47] A dictionary of commonly occurring visual keypoints is constructed through clustering and for each image the number of occurrences of each of these keypoints is counted. The size of the resulting feature vector depends on the number of common keypoints that are in the dictionary and is typically chosen at one hundred.

3.1.2 Two-point statistic features

A more physically inspired approach to extract features from microstructure images is based on the two-point statistic. The starting point for computing the two-point statistic, is the microstructure function $m(\mathbf{r}, n)$, which represents the probability of finding the local state n at spatial location $\mathbf{r}$.[48] For our purposes, the local state will always refer to which steel phase is present at location $\mathbf{r}$. As we are considering images, the spatial locations are discretized on a uniform grid. It is convenient to introduce the notation m_s^h for the discretized microstructure function, where $h = 1..H$ refers to one of the H different phases and $s = 1..S$ refers to one of the S different grid points. As m_s^h represents a probability function, it satisfies the following properties

$$\sum_{h=1}^{H} m_s^h = 1, \qquad 0 \leq m_s^h \leq 1.$$

The two-point correlations are defined as[49]

$$f_s^{hh'} = \frac{1}{S} \sum_{r=1}^{S} m_r^h m_{r+s}^{h'}.$$

For microstructures in which a lot of different phases are present, many two-point correlations can be computed and the representation of the microstructure becomes highly unwieldy. Fortunately, it can be proven that for a microstructure containing H different phases, there are only $H - 1$ independent two-point correlations, from which the remaining ones can be computed. Hence, only one two-point correlation is required for dual phase steels. Because of this, the two-point correlation is mainly used for describing dual phase steels. Even then, the dimensionality of the two-point correlation is still S, which for image data amounts to the number of pixels in the image. This is still too much to yield a practical representation of the microstructure. To further reduce the dimensionality of the representation, Principal Component Analysis (PCA)[50] is used. The two-point correlation function is written in the basis of eigenfunctions ϕ_{is} of the covariance matrix, the so-called principal components

$$f_s^{hh'} \approx \sum_{i=1}^{R} \alpha_i^{hh'} \phi_{is} + \bar{f}_s, \tag{3.1}$$

where $\alpha_i^{hh'}$ are the coefficients in the principal component basis and $\bar{f}_s$ is the average of the two-point correlation for the entire dataset of microstructures at grid point s. The approximation in equation (3.1) is because only the R

principal components with the highest eigenvalues have been retained. Thus, the microstructure of a dual phase steel can be represented in R numbers $\alpha_i^{hh'}$, where the value of R can be freely chosen. This type of representation has already been studied in the context of structure-property prediction.[51]

Other physically inspired approaches to represent the microstructure are based on morphological parameters of the phase grains that are visible in the image.[52] Examples of such parameters are the area, the perimeter and the shape factor of the grain.

3.2 Deep learning

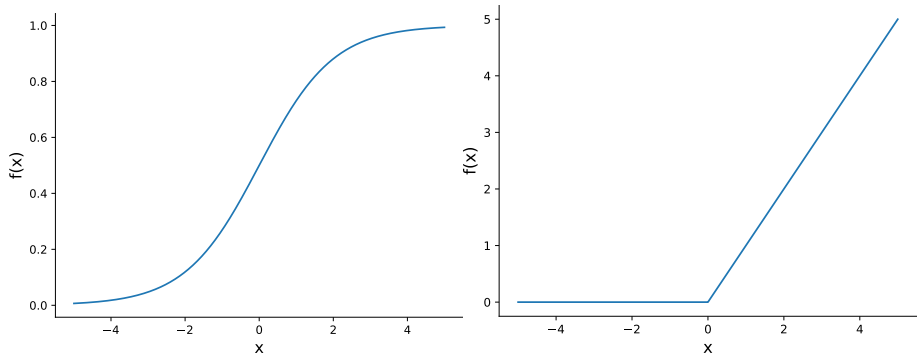
In 2012 the public interest in deep learning started to take off. On the one hand, there was the publication of Ciresan *et al.*[53] who demonstrated that deep learning could obtain superhuman performance in traffic sign recognition. On the other hand, there was Alex Krizhevsky *et al.*[54] who won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) with a deep learning model, outperforming all other machine learning methods. From that point onwards it became clear that deep learning holds the potential to outperform humans in the analysis of visual data. Nowadays, deep learning has completely dominated the scene of computer vision and other machine learning methods are only rarely used when dealing with visual data. As deep learning plays a crucial role in this work, we discuss the fundamentals here.

A deep learning model, also called a deep learning network, can be thought of as a lasagna, which contains many layers, each with its specific function. Those layers constitute the atomic building blocks of deep neural networks. We therefore discuss them first. Then, we explain how deep learning networks are trained. Lastly, we discuss the main network architectures used in this work.

3.2.1 Layers

Conceptually, the simplest layer is the **densely connected layer**, which corresponds to a matrix-vector multiplication. Given a N-dimensional feature vector $\mathbf{x}$, a $M \times N$ weight matrix W, a M-dimensional bias vector $\mathbf{b}$ and a non-linear activation function f, the M-dimensional output vector $\mathbf{y}$ of the dense layer is computed as

$$\mathbf{y}_i = f\left(\sum_{j=1}^{N} W_{ij}\mathbf{x}_j + \mathbf{b}_i\right).$$



(a) sigmoid **(b)** ReLU

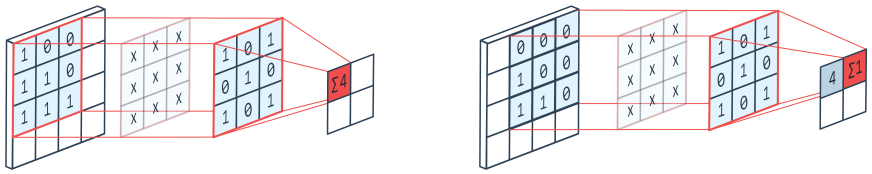
Figure 3.1: Two common types of activation functions.

The elements of the weight matrix W and the bias vector $\mathbf{b}$ are the model parameters that have to be optimized based on the task at hand. Before the advent of deep learning, densely connected layers were already commonly used in regular neural networks. By a regular neural network, we mean a network that only has one to three layers of trainable parameters, whereas deep learning networks nowadays can easily have more than fifty layers. The activation function f can be any differentiable function, but the two types of functions shown in Figure 3.1 are the most commonly used. S-shaped functions like the sigmoid function shown in Figure 3.1a were commonly used in regular neural networks, but are not as commonly used in deep neural networks because of difficulties in the optimization. The class of Rectifying Linear Unit (ReLu) activation functions is currently much more popular.

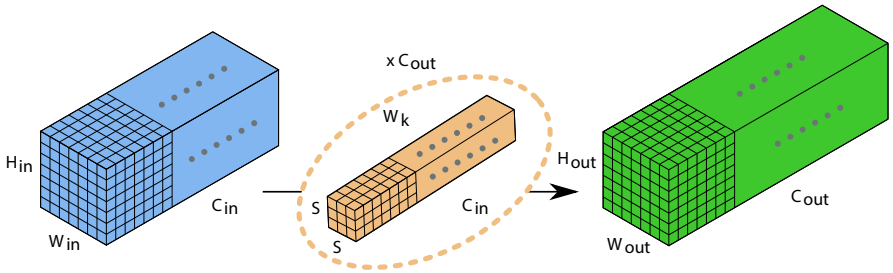
The disadvantage of a densely connected layer is that the product $M \times N$ can become very big when dealing with high-dimensional feature vectors. For instance, when dealing with an image containing 200×200 pixels each with three color channels, the dimensionality of the feature space is already 120 000. Given that M will most likely be of the same order of magnitude, this results in more than a billion trainable parameters for a single layer. This is unfeasible, both in terms of memory and computational time. The key breakthrough in deep learning for computer vision was the introduction of the **convolutional layer**[55]:

$$(y_k)_{ij} = f\left((W_k * X)_{ij} + \mathbf{b}_k\right).$$

Here, X is a $C_{in} \times H_{in} \times W_{in}$ dimensional tensor, with C_{in} denoting the number of pixel channels and H_{in} and W_{in} denoting the spatial dimensions. The index $k = 1...C_{out}$ indicates the k-th channel of the convolution with



(a) spatial convolution.



(b) the convolutional layer.

Figure 3.2: (a) An illustration of how the spatial convolution works. Image taken from [56]. (b) A visual explanation of how a convolutional layer works. By applying C_{out} different spatial convolutions, an output tensor with a different number of channels can be obtained.

C_{out} the number of channels in the output. The indices i and j indicate the positions along the spatial dimensions. The asterisk denotes the spatial convolution operator, which is illustrated in Figure 3.2a. The convolutional tensor W_k has size $C_{in} \times S \times S$ with S denoting the size of the kernel of the convolution. As is illustrated in Figure 3.2b, a convolutional layer applies C_{out} spatial convolutions simultaneously. The different convolutional tensors can be grouped in a single convolutional tensor W of size $C_{out} \times C_{in} \times S \times S$. All elements of this tensor are trainable parameters of the convolutional layer. The other trainable parameters are the elements of the bias vector $\mathbf{b}$, which is C_{out}-dimensional. Zero padding is commonly applied so that usually $H_{in} = H_{out}$ and $W_{in} = W_{out}$. The usage of spatial convolutions in image processing was not new, as it was already commonly used in filters to for instance sharpen or blur images. However, the main advantage of the convolutional layer is that the size of neither the tensor W nor $\mathbf{b}$ depend on the size of the input and the values C_{out} and S can be freely chosen. In Figure 3.2a, we show a convolution with $C_{out} = 1$ and $S = 3$. In a typical setting, C_{in} and C_{out} can be up to the order thousands, whereas S will be either three or five. Thus, the dimensionality of W will be up to the

order of millions, rather than billions irrespective of the size of the image. This makes it feasible to build networks with many layers. Compared to the densely connected layer, where the interaction between all input features is taken into account, it might look as if the convolutional layer is much less expressive, as the convolutional layer restricts the interaction to the neighbouring pixels. However, by stacking many convolutional layers, it is possible to obtain interactions between pixels that are remote from each other. This approach turns out to be much more effective for computer vision tasks than using densely connected layers.

One of the problems with stacking many layers, is that the variance of the output of the intermediate layers can become either very big or very small. In both cases, this results in an ill-conditioned optimization problem. To deal with this, Ioffe & Szegedy[57] introduced the **batch-normalization layer**. This type of layer leverages the fact that a deep neural network is typically trained iteratively on small subsets of the training data, as we will explain in more detail in section 3.2.2. These subsets are called batches. Batch-normalization layers can be employed for both vector data and higher order tensors with only minor modifications. For notational simplicity, we discuss the case where the input of the layer is a vector. If we denote X a $M \times N$ dimensional matrix, with M the number of samples in the batch and N the size of the input vectors, a batch-normalization layer performs the following computations

$$\begin{aligned}
\boldsymbol{\mu}_B &= \frac{1}{M}\sum_{i=1}^{M} X_i \\
\boldsymbol{\sigma}_B^2 &= \frac{1}{M}\sum_{i=1}^{M}(X_i - \boldsymbol{\mu}_B)^2 \\
\hat{X}_i &= \frac{X_i - \boldsymbol{\mu}_B}{\sqrt{\boldsymbol{\sigma}_B^2 + \epsilon}} \\
Y_i &= \boldsymbol{\gamma} \odot \hat{X}_i + \boldsymbol{\beta},
\end{aligned}$$

where $\boldsymbol{\mu}_B$ and $\boldsymbol{\sigma}_B$ are N-dimensional vectors representing the mean and standard deviation of the batch. The ϵ is a small positive number that assures numerical stability. The "$\odot$" represents the element-wise multiplication. X_i and Y_i are respectively the input and output of the i-th sample in the batch. The output matrix Y thus has the same size as the input matrix X. $\boldsymbol{\mu}_B$ and $\boldsymbol{\sigma}_B$ are typically computed as running statistics over the different batches. The intermediate input is standardized using these batch statistics, so that it has approximately zero mean and unit variance. After this standardization, the output is obtained by rescaling the input by a factor $\boldsymbol{\gamma}$ and adding an

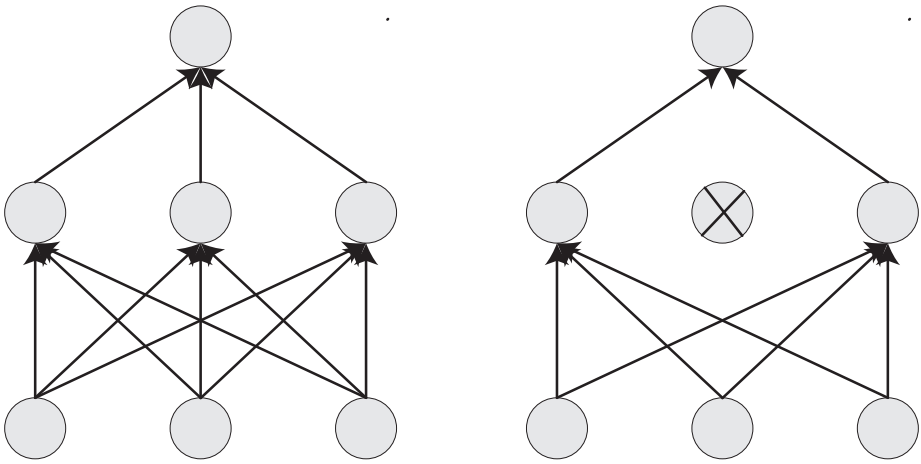


Figure 3.3: An illustration of how a dropout layer works.

offset $\boldsymbol{\beta}$. Both $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are N-dimensional vectors and are the trainable parameters of the layer. While batch-normalization layers are conceptually relatively simple, they have played a major role in enabling the effective training of deep neural networks. For image data, where the input X has the shape $C \times H \times W$, the batch statistics are computed per channel, so that $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are C-dimensional vectors in that case. This results once more in feasible amount of trainable parameters that is irrespective of the spatial dimensions of the input image.

Despite the usage of convolutional and batch-normalization layers, the total number of parameters in a deep neural network can easily amount to a few million. This is a very large number and is not easy to guarantee that models with such a large amount of trainable parameters can robustly fit the data. One way to improve the robustness of the model predictions, is by introducing **dropout layers**[58]

$$\begin{aligned} \boldsymbol{r}_i &\sim Bernoulli(p) \\ \boldsymbol{y}_i &= \frac{1}{p}\boldsymbol{r}_i * \boldsymbol{x}_i, \end{aligned} \tag{3.2}$$

where $\boldsymbol{r}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are all N-dimensional vectors. Dropout layers randomly put intermediate inputs equal to zero with a probability of $1 - p$, as is illustrated in Figure 3.3. This results in less correlated features and leads to more robust model predictions. The factor $\frac{1}{p}$ in equation (3.2) assures that the sum of the features remains the same on average. The disadvantage of dropout layers is that the random omission of certain features results in a more complicated optimization problem, which leads to longer training times.

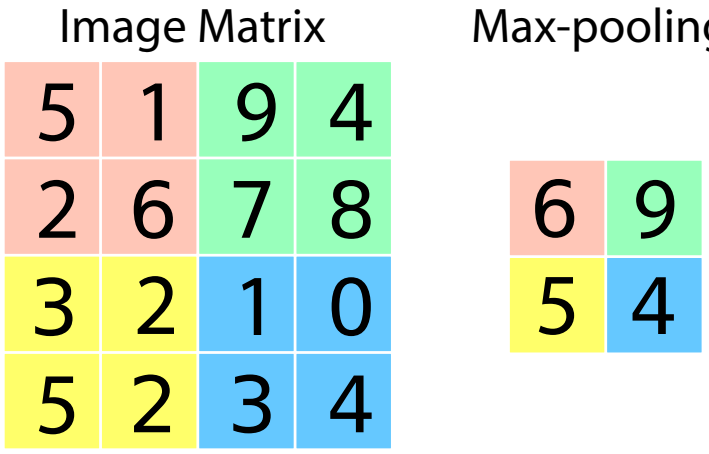


Figure 3.4: An illustration of how a max-pooling layer works.

While convolutional layers permit to deal with image data efficiently, it is for many applications useful to eventually convert the image into a vector. The typical approach to do so, is by gradually decreasing the spatial dimensions of the intermediate outputs throughout the neural network while increasing the number of channels. The latter can be achieved by using convolution layers for which $C_{out} \geq C_{in}$. Once the number of spatial dimensions has become sufficiently small, it is possible to compute the average or the maximum over the spatial dimensions to obtain a feature vector with the number of features equal to the number of channels before averaging or before taking the maximum. Such layers are called **global average pooling layers** or **global max pooling layers**, respectively.[59] The term global refers to the fact that these operations can be carried out regardless of the exact spatial dimensions of the image. It is also possible to apply both global average and global max pooling and to concatenate the resulting feature vectors.

This brings us to the following question: how can we reduce the number of spatial dimensions efficiently? In earlier network architectures, **average pooling or max-pooling layers** were commonly used. In Figure 3.4, we illustrate how such a max-pooling works. For a given kernel size, 2×2 in case of the illustration, the maximum element is computed per subregion. Only the maximum element is retained, thus reducing the number of spatial dimensions. The max-pooling operation shown in the figure is very similar to how a convolution works. The main difference is that the pooling operation is applied per subregion rather than per pixel. However, for convolutions it is also possible to move the kernel with a step size that is bigger than one pixel. For instance in the illustration, we could move the kernel by two pixels

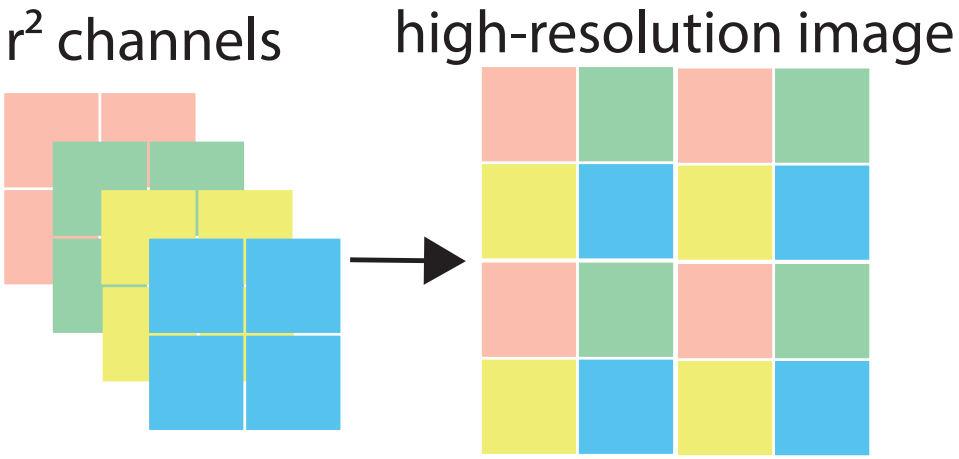


Figure 3.5: An illustration of how a pixel shuffle layer works[60].

at the time and the resulting image would have the same shape as the one obtained by max pooling. Such a convolution is called a strided convolution and the stride refers to the step size that is taken expressed in number of pixels. Thus, it is also possible to reduce the spatial dimensions through the use of **strided convolutions**. This is the most commonly used approach nowadays as it gives more flexibility than conventional pooling layers.

A last operation that is useful for certain applications is the unpooling layer or the upsampling layer. This type of layer can be considered as the inverse operation of a pooling layer. Its aim is to increase the spatial dimensions of the intermediate inputs in the network. While many different approaches exist to construct such unpooling layers, we only mention the **pixel shuffle layer**[60]. The working of this layer is illustrated in Figure 3.5. As the name implies a pixel shuffle layer reshuffles the dimensions of an image so that a $r^2C \times H \times W$ sized image turns into a $C \times rH \times rW$. This increases the spatial resolution by a factor of r.

3.2.2 Optimization of a deep neural network

A deep neural network can easily contain several millions of trainable parameters. Therefore, the optimization of those parameters is not a trivial task. For a given image X, model parameters $\boldsymbol{\theta}$, the output of the network is given by:

$$\hat{\mathbf{y}} = f_\theta(X),$$

as the deep neural network can be seen as a high-dimensional function that takes as input an image and returns a prediction for the task at hand. For a

microstructure recognition problem, this prediction can be a probability that the material on the image is for instance martensite, bainite or austenite, while for regression the prediction can for example be the estimated hardness of the material on the image. For a given loss function L and target value $\mathbf{y}$, the optimization of the network can then be written as

$$\arg\min_{\theta} L(\mathbf{y}, \hat{\mathbf{y}}).$$

The most commonly used loss function for classification is the cross-entropy loss[61], defined as

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{N} \mathbf{y}_i \log(\hat{\mathbf{y}}_i), \tag{3.3}$$

here $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ represent the true and predicted probability for class i of the N predefined classes in the classification problem. Most commonly, $\mathbf{y}_i$ equals one if the material belongs to class i and is zero otherwise. Since we consider $\hat{\mathbf{y}}$ to be a probability distribution, the cross-entropy then boils down to maximizing the log-likelihood of the correct class.

For regression, the most commonly used loss function is the mean squared error (MSE)[61]:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2N} \sum_{i=1}^{N} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2, \tag{3.4}$$

which is also commonly used in linear regression. Here, N is number of components of the vector $\mathbf{y}$.

Now that we understand what we aim to optimize, we can focus on how to optimize this. The parameters of a deep learning network are mostly optimized through gradient descent. However, this requires to be able to compute the gradients with respect to all parameters in the network. These can be obtained through backpropagation[62], which is essentially a different name for the chain rule in analytical calculus. In Figure 3.6 we illustrate how backpropagation works for a layer l. We assume that $\mathbf{i}_l$, $\mathbf{o}_l$ and $\boldsymbol{\delta}_l$ have already been computed either during the forward pass or during the backpropagation in later layers. Given these values, we can compute the gradient of the loss L with respect to the layers parameters $\boldsymbol{\theta}_l$ as

$$\frac{\partial L}{\partial \boldsymbol{\theta}_l} = \frac{\partial L}{\partial \mathbf{o}_l} \frac{\partial \mathbf{o}_l}{\partial \boldsymbol{\theta}_l} = \boldsymbol{\delta}_l \frac{\partial \mathbf{o}_l}{\partial \boldsymbol{\theta}_l},$$

as δ_l is already computed and $\frac{\partial \mathbf{o}_l}{\partial \boldsymbol{\theta}_l}$ can be readily computed for layer l, the derivative can be straightforwardly evaluated. For the computation of the

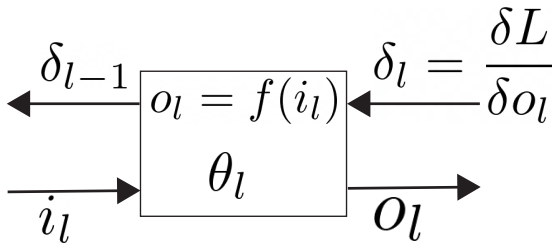


Figure 3.6: The backpropagation mechanism for a given layer l with parameters θ_l.

gradient in earlier layers, we need to compute δ_{l-1}, which can be done as

$$\delta_{l-1} = \frac{\partial L}{\partial \mathbf{o}_{l-1}} = \frac{\partial L}{\partial \mathbf{i}_l} = \frac{\partial L}{\partial \mathbf{o}_l}\frac{\partial \mathbf{o}_l}{\partial \mathbf{i}_l} = \boldsymbol{\delta}_l \frac{\partial \mathbf{o}_l}{\partial \mathbf{i}_l},$$

where we relied on the fact that $\mathbf{o}_{l-1} = \mathbf{i}_l$. The factor $\frac{\partial \mathbf{o}_l}{\partial \mathbf{i}_l}$ can again be computed for layer l. By repeating this procedure backwards we can obtain the gradients with respect to all parameters in the neural network. Note that the gradients of layer l are proportional to the product of the gradients of all later layers. This can lead to problems with the optimization as we will discuss later.

Once the gradients are computed, we can update the parameters in the model as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\partial L}{\partial \boldsymbol{\theta}}, \tag{3.5}$$

where the index t refers to the iteration number and η is the learning rate. A high learning rate will result in fast changing model parameters. This might lead to faster convergence, but it can also result in numerical instabilities. The learning rate can also change as a function of the iteration number and many different learning rate schedule have been proposed in the literature.

As was mentioned before, it is most of the times unfeasible to compute the gradients for all the training data simultaneously. Therefore, the training data is split into smaller subsets which are called batches. The model is shown one batch per iteration and its parameters are iteratively updated based on the gradients computed using the samples in the batch. By computing gradients on more than one sample at the same time, the gradients become more representative leading to a faster and more stable optimization of the network. An iteration refers to one update of the model parameters, whereas a complete pass through all the training data is called an epoch. The number of samples in the batch, commonly called the batch size, plays an important role. A bigger batch size results in more representative gradients and more stable optimization. However, a smaller batch size implies

more updates of the parameters in one epoch and thus potentially a faster optimization. However, the gradients are noisier which might lead to slower convergence.

The update rule shown in equation (3.5) is called Stochastic gradient descent (SGD)[61]. The term stochastic refers to the fact that the batches are randomly selected, thus introducing a stochastic component in the optimization procedure. SGD can be sped up by including a momentum term[63]:

$$\mathbf{v}_{t+1} = \mu \mathbf{v}_t - \eta \frac{\partial L}{\partial \boldsymbol{\theta}}$$
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \mathbf{v}_{t+1},$$

where $\mathbf{v}_t$ is the velocity at iteration t and μ is the momentum parameter. The velocity can be seen as a weighted average over gradients from the current and previous iterations. This can help to stabilize the training and speed up the convergence. An alternative to the regular momentum, is the Nesterov momentum[63], defined as:

$$\mathbf{v}_{t+1} = \mu \mathbf{v}_t - \eta \frac{\partial L}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}_t + \mu \boldsymbol{v}_t}$$
$$\theta_{t+1} = \boldsymbol{\theta}_t + \mathbf{v}_{t+1},$$

where the gradient is now evaluated at $\boldsymbol{\theta}_t + \mu \mathbf{v}_t$ rather than at $\boldsymbol{\theta}_t$. Compared to regular momentum, Nesterov momentum is more stable, as it prevents the momentum term from overshooting or going into the wrong direction.

A potential disadvantage of SGD is that the same learning rate is used for all parameters in the network. Adaptive learning rate methods permit to use different learning rates for different parameters. The most commonly used adaptive optimizer is Adam[64]. Adam keeps track of the decaying averages of both the first and second moment of the parameters as

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \left(\frac{\partial L}{\partial \boldsymbol{\theta}} \right)_t$$
$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \left(\frac{\partial L}{\partial \boldsymbol{\theta}} \right)_t^2,$$

where $\mathbf{m}_0$ and $\mathbf{v}_0$ are set to zero. The β_1 and β_2 are the decay rates and are kept fixed. The default value proposed by the authors for β_1 is 0.9 and for 0.999 for β_2. Because of the initial values of $\mathbf{m}_0$ and $\mathbf{v}_0$, the unbiased estimates of the first two moments are given by

$$\hat{\boldsymbol{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t}$$
$$\hat{\boldsymbol{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t}.$$

The parameters are updates by the following rule

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon}\hat{\boldsymbol{m}}_t, \tag{3.6}$$

where ϵ is a small positive number. It is clear that the effective learning rate in equation (3.6) now depends on the second moment of the parameters.

3.2.3 Initialization of the model parameters

From our discussion about backpropagation, it is clear that the gradients of the parameters in layer l are proportional to the product of the gradients of all later layers. This poses a potential problem for very deep neural network, as too small gradients will cause the so-called vanishing gradient problem. The gradients become so small that the parameters no longer significantly change during optimization. On the other hand, if the gradients are too big, this will lead to exploding gradients causing convergence problems and potentially numerical issues. To deal with the problem of vanishing and exploding gradients, several improvements have already been made. Two improvements have already been briefly discussed: the use of batch-normalisation layers and the use of non-sigmoid activation functions are highly beneficial for keeping the size of the gradients in an acceptable range. A third improvement is changing the initialization of the parameters.

He *et al.*[65] demonstrate that for ReLU activation functions and under the assumptions that both inputs and the parameters are independent and identically distributed and that the inputs and the parameters are mutually independent, the variance of the outputs of all intermediate layers will be the same if the parameters of layer l are initialized as

$$\boldsymbol{\theta}_l \sim \mathcal{N}\left(0, \frac{2}{n_l}\right).$$

Here, n_l can be either the number of inputs features or the number of output features of the layer. The harmonic mean of the number of input and output features is also sometimes used.

Many other initialization schemes exist. A well-known alternative to He initialization is Glorot Initialization[66]. This type of initialization is derived using similar assumptions as He made, but for sigmoid-like activation functions rather than rectifiers.

A completely different approach to initialize the parameters of the model is by taking the parameters of a similar model that has been trained on a different task. This is called transfer learning[67], as it transfers the

knowledge of the network acquired on one task to a new task. The network that has been trained on the first task is called the pretrained network. In practice, almost all pretrained networks are obtained from a classification task on the very popular and publicly available ImageNet dataset[68], which contains more than fourteen million images of all kinds of categories such as animals, vehicles and people. Transfer learning is very commonly used nowadays because it permits to train deep neural networks on much smaller datasets. Because the initial values of the parameters of the network typically already achieve good performance on new tasks, only some fine-tuning of the parameters is required. When using transfer learning, one therefore preferentially uses a smaller learning rate than when training a network from scratch. The main limitation of transfer learning is that it can only be used if the model architecture is completely the same. In practice, this is not a heavy restriction as only a limited set of model architectures are commonly used. In the next section, we discuss some popular networks architectures that are relevant for this work.

3.2.4 Commonly used architectures

In this section, we discuss the main deep learning networks that were used in this work. In Figure 3.7, we show the VGG16 network[69] applied to a classification task with K predefined classes. This network obtained the first place in the ImageNet Large Scale Visual Recognition Competition[54] of 2014 in the localisation track and a second place in the classification track, implying that in 2014 this network could be considered state-of-the-art. The network architecture is relatively simple with blocks consisting of convolutions and ReLU activations functions. At the end of every block, a max-pooling operation is applied to reduce the spatial dimensions. At the same time, the number of features channels is increased by employing convolutional layers that have a higher number of output channels than the number of input channels. Once the number of spatial dimensions is sufficiently small, the tensor data is reshaped into a vector. This vector is then used as input for a dense layer to carry out the classification. The first five blocks have the function of extracting features from the image and serve to convert the image into a vector representation. The original VGG16 architecture had 16 layers with trainable parameters. In 2014, this was enough for the authors to call their network "very deep". Nowadays, networks of up to thousand layers can be trained thanks to improvements in initializations and batch-normalization layers and changes in architecture.

One of the most important changes in the network architecture was introduced in 2015 by He *et al.*[70]. They introduced what is called residual

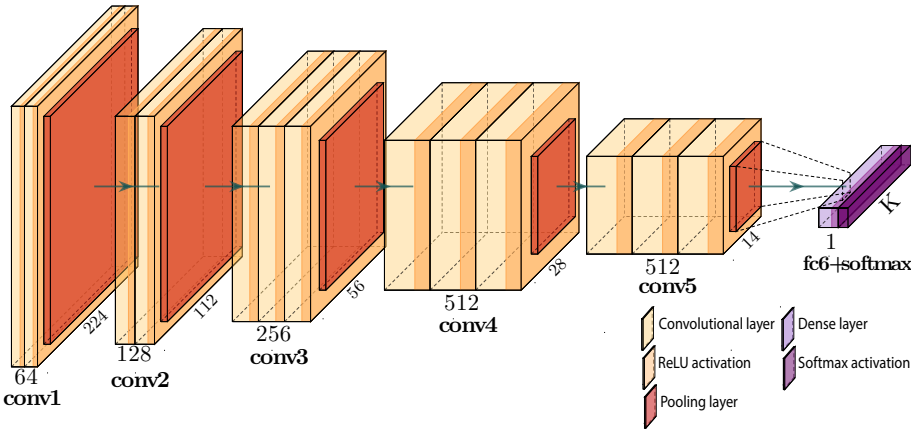


Figure 3.7: A schematic illustration of the VGG16 architecture. For clarity, we have reduced the number of dense layers, since the original architecture contained three dense layers at the end.

networks (ResNets). These models rely heavily on residual blocks like the one shown in Figure 3.8. In these blocks, the input is added to the output of the block through a shortcut connection. While this may not seem like a big change, it allowed the authors of the paper to train models with more than 100 layers with trainable parameters, where before it was difficult to go beyond 20 layers. The introduction of the residual block therefore enables network architecture to become deeper and better performing. The reason why a simple shortcut connection makes so much difference, is that additional layers can simply be ignored by the network if it learns the identical function $F(x) = 0$. Learning such a function is much easier for a neural network than learning the identity operation, which would be required for networks without shortcut connections. The $F(x)$ can therefore be considered as a perturbation to the input.

In Figure 3.8 it is not clear how the trainable layers are defined. The original ResNet paper proposed a simple stacking of convolutional layers as is shown in Figure 3.9 (left). In a later paper, Xie *et al.*[71] proposed to replace the single convolution by a set of parallel convolutions as is shown in Figure 3.9 (right). This type of model architecture is called ResNeXt. By introducing many parallel convolutions, the authors claim that it is possible to obtain better performing models for the same level of complexity. Both the ResNet50 and ResNeXt50 architecture use a global average pooling layer as a final layer to obtain a 2048-dimensional feature vector.

A last type of architecture that is relevant for this work is the Unet[72], which is shown in Figure 3.10. Unlike the previous architectures, this type of model

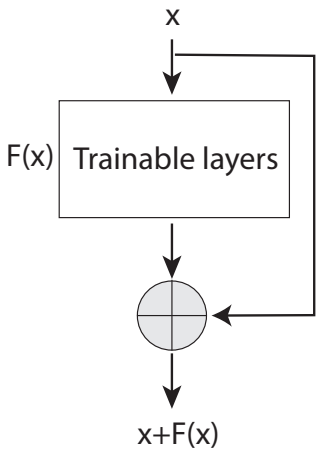


Figure 3.8: A schematic illustration of the residual block introduced by He *et al.*[70].

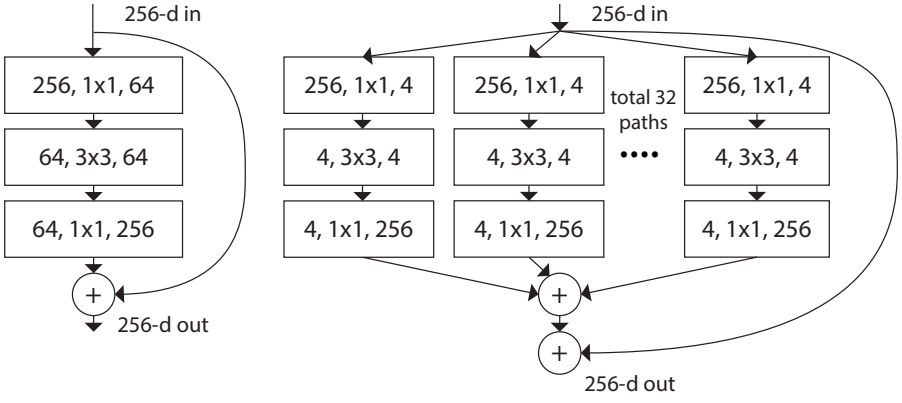


Figure 3.9: The elementary building block for a ResNet (left) and a ResNeXt (right) architecture. Image taken from [71].

does not convert the image into a vector. Instead, it is used for tasks that require pixel-level output. An example of such a task is shown in the figure. The model takes an image as input and outputs pixel-level predictions about the steel phase. Such a task is called image segmentation and Unets have already been applied to this task in the context of microstructure segmentation.[73] In this work we will not use Unets for segmentation tasks, but for artificially enhancing the spatial resolution of images.

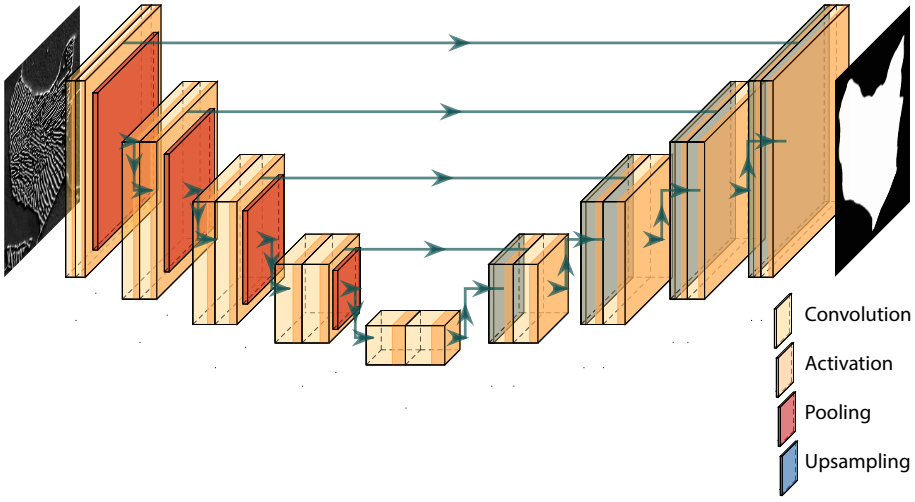


Figure 3.10: A schematic illustration of the Unet architecture used for a segmentation task.

4

Representing a microstructure using deep learning

The key to artificial intelligence has always been the representation.
Jeff Hawkins, AI entrepreneur

4.1 Introduction

The question of how a microstructure can be represented in a limited set of numbers or descriptors has been studied by many metallurgists and statisticians in the past decades.[74] This question is highly relevant, because in order to quantitatively analyse a microstructure it is inconvenient to work with raw image data. Most microscopy images of microstructures easily have over a million pixels and each of these pixels is a degree of freedom. If we want to establish PSP links, a much more low-dimensional or compact representation is needed. In the literature, many types of representations have been tried. Some rely on physical insight, such as the two-point statistic.[49] Others are based on classical computer vision methods, such as the Haralick features[45, 46] or keypoints-based approaches[47]. Representations based on pretrained deep learning models have also been used.[75] However, none of these approaches leverages one of the key strengths of deep learning: the ability to learn representations purely based on data. In this chapter, we discuss a principled approach to learn compact, yet highly

informative representations of microstructural images using a method called triplet networks.[76] We will first discuss the methodology and the dataset, before we summarise the main results.

4.2 Methodology

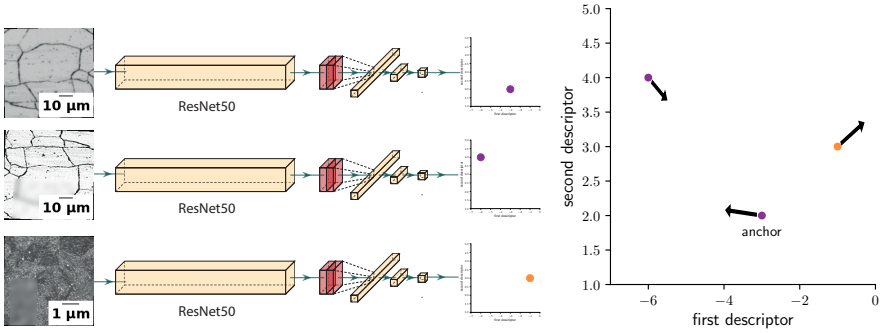
Most deep learning architectures first convert image data into a feature vector and then use this vector to perform a certain task such as regression or classification. However, it is possible to only focus on the conversion of the image to a vector. Specifically, we have a network f with trainable parameters $\boldsymbol{\theta}$ that takes as input an image X and returns a vector $\mathbf{y}$

$$\boldsymbol{y} = \boldsymbol{f_\theta(X)}. \tag{4.1}$$

Rather than showing only one image at the time to the network, we show three images simultaneously. The first image X_a is the reference image or the anchor. The second image X_p is a positive example that belongs to the same material as the anchor. The last image X_n is a negative example and depicts a different material. The triplet loss[77] is defined as

$$\max\left(0, ||f_\theta(X_a) - f_\theta(X_p)|| - ||f_\theta(X_a) - f_\theta(X_n)| + \alpha\right),$$

where $||.||$ represents the euclidean norm and α is a positive number called the margin that represents the desired distance between the representations of different materials. A visual explanation of how the triplet network works, is given in Figure 4.1. In the forward pass (a), the representations of the images are computed. The two upper images belong to the same material, whereas the lower image belongs to a different material, as is reflected in the colour of the dots. During the backward pass (b), the parameters of the neural network are updated so that the representation of the anchor moves towards the other representation of the same material and away from the representation of the other material. This entire procedure is repeated over 100 000 times on different triplets of images. Thus, the deep learning network learns to represent microstructure images in a very compact manner. The advantages of a triplet network is twofold. First, the representations are obtained by optimizing a distance-based loss function, as is given in equation (4.2). The distance can therefore be seen as a similarity measure: points laying close to each other represent materials that have a visually similar microstructure. A second advantage is the flexibility in choosing the dimensionality of the representations. In the figure, the network returns a two-dimensional representation, but we are free to select any dimensionality we desire. However, we would prefer low-dimensional representations, as



(a) forward pass **(b)** backward pass

Figure 4.1: Illustration of the training procedure of the triplet network. Image adapted from [79].

they can be easily visualised and because machine learning models with a limited amount of inputs tend to be more robust.[78] The latter is useful if we want to use the obtained representations as input for a machine learning model.

The main difference between the network architecture of a triplet network and a convolutional neural network (CNN) used for image classification is that the latter has an additional densely connected layer that takes as input the representation vector $\mathbf{y}$ in equation (4.1) and returns a C-dimensional vector $\mathbf{z}$ with C the number of classes in the classification problem. A softmax function is then applied to the vector $\mathbf{z}$:

$$\mathbf{p}_i = \sigma(\mathbf{z})_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{C} e^{\mathbf{z}_j}}.$$

This ensures that the elements $\mathbf{p}_i$ of vector $\mathbf{p}$ are positive and sum to one. The elements $\mathbf{p}_i$ can be interpreted as the probability of the image belonging to class i. The model is optimised using the cross-entropy loss, which we defined in equation (3.3). Applying the densely connected layer in combination with the softmax function to the representations $\mathbf{y}$ is the same as if we would apply logistic regression to $\mathbf{y}$. The representations of this type of CNN are therefore optimised to be linearly separable. Because of the last densely connected layer, the CNN used for classification has parameters that are class-specific. This makes such type of networks less feasible to deal with problems where there are very few examples for each class. Such problems are called one-shot or few-shot learning problems and for this type of problems triplet networks are preferred.

We use a ResNeXt50[71] architecture to convert the image into a feature vector of a fixed size of 4096. This feature vector is the concatenation of the 2048-dimensional output of a global average pooling layer and the 2048-dimensional output of a global max pooling layer. The ResNeXt50 is pretrained on the ImageNet dataset.[68] Some additional densely connected layers are added to the network to reduce the dimensionality of the representations. These layers are trained from scratch and we are therefore free to choose the dimensionality of the final representations. We use a two-stage procedure, in which we first keep the pretrained parameters fixed and only train the parameters of the densely connected layers. After this, we fine-tune all parameters jointly at a significantly lower learning rate. All optimization are performed using the SGD optimizer.

A last important ingredient for the study of microstructure representations is a metric in order to be able to compare different representations with each other. We propose to tackle a microstructure recognition problem, where based on the representations, a machine learning model should be able to recognise which image belongs to which material. The metric we use is the classification accuracy, defined as

$$accuracy = \frac{\#\ images\ correctly\ classsified}{\#\ images\ classsified}.$$

The rationale behind this metric is that a good representation should enable at least to tell the difference between different materials. Whether this criterion is sufficient to establish robust PSP-links is unlikely, but is definitely a necessary condition. The machine learning model we train is a random forest classifier.[80] There are several reasons why we prefer this type of classifier. First, it is robust to the choice of hyperparameters and we found in our experiments that the default values typically yield good results. This eliminates the need for a validation set, so that we only have to split the data into a training set to train the model and a test set to evaluate the model performance. Second, it permits to compute the out-of-bag score on the training data. This score is a good indication for the generalization performance of the classifier, giving an additional indication of how general our models are. Lastly, we found that random forests work well both in low- and high-dimensional input spaces. This is important, as we intend to compare diverse representations of a different dimensionality.

4.3 Datasets

Two datasets are used in this work. The first dataset consists of optical microscopy images of five clearly different material groups, as is shown in

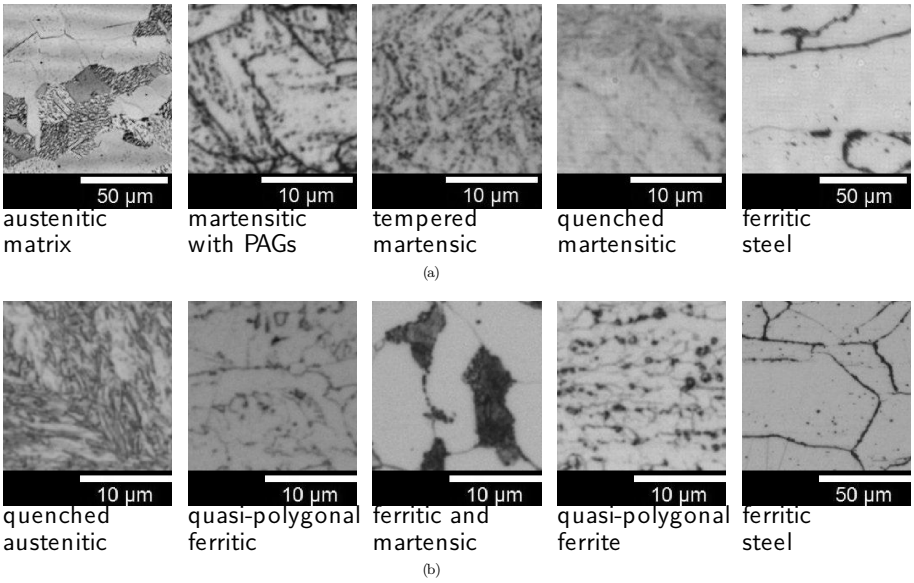


Figure 4.2: Some examples of the crops that are used to train the deep learning models for both dataset 1 (a) and dataset 2 (b). Figure taken from [81].

id	description	etching	# materials	colourmap
1	austenitic matrix	Nital	6	grey
2	martensitic with PAGs	Bechet-Beaujard	21	orange
3	tempered martensitic	Nital	19	blue
4	quenched martensitic	Nital	9	green
5	ferritic steels	Nital	5	purple

Table 4.1: Description of the different groups of materials in the first dataset and the number of different material classes per group. We attribute a colourmap to every material group for later figures. Table adapted from [81].

Table 4.1. For each material group, there are a number of materials that differ from each other in terms of processing and composition. In the literature, most research focuses on recognising the right material group.[45–47, 75, 82] Our aim is to go beyond this and we want to investigate to which extent the different representations are able to distinguish between materials with only small differences in processing and composition. Such materials are visually very similar, so that even for expert metallurgists it is not always possible to recognise the correct material based on a single image. In total, the first dataset contains 60 different material classes. By defining classes based on processing and composition, we use a quantitative criterion to differ between the classes and ensure that every image depicts only one material class. For each material, we have 7 to 24 optical microscopy images with a 1000×1200 pixel resolution. The magnifications, expressed as inter-pixel distances, range from 0.1 μm to 5 μm. To obtain a training and test set, we take 80 % of the images as training set and the remaining 20 % as test set. For the training set, we randomly generate 500 crops from the selected images with a pixel resolution of 200×200 for each material. For the test set, we generate 125 crops with the same pixel resolution for each material. This procedure is repeated three times to avoid dependence on the precise split of training and test set. It is important to stress that different images are used in the training and test set in order to allow an unbiased evaluation of the representations.

class id	description
1	quenched martensite
2	quasi-polygonal ferrite
3	ferrite + pearlite
4	quasi-polygonal ferrite
5	ferritic steel
6	quasi-polygonal ferrite
7	austenite + pearlite
8	quasi-polygonal ferrite
9	tempered martensite
10	granular bainite

Table 4.2: A description of each of the material classes in dataset 2. Table adapted from [81].

The second dataset consists of 30 optical microscopy images with a pixel resolution of 1000×1200 belonging to 10 visually clearly different materials. For each material there are three images in the dataset at a fixed magnification. The magnifications over all materials ranges from 0.1 μm to 0.5 μm. The dataset is solely used to assess how well the representations obtained

by training a network on the first dataset generalize to the new materials in the second dataset. In order to split the data into a training and a test set, we use three-fold cross-validation, where one image is put in the test set and the remaining two are used to train the random forest classifier. Some example images of both datasets are shown in Figure 4.2. More examples and information on the datasets can be found in the reference paper.[81]

4.4 Results

Figure 4.3 summarises the main result of our study. Figure 4.3 (b) shows the two-dimensional representations of the test set of dataset one obtained by the triplet network. Material groups all have the same colour, as is indicated in Table 4.1, while the dots belonging to same material have the same brightness. We see that the representations of images belonging to the same material are clustered together and relatively well separated from the representations of other materials. For the representations of the CNN classifier, which are shown in Figure 4.3, we see that they are not lying as closely together, but rather form elongated lines. This can be explained by remembering that the requirement for the representations of the CNN classifier is that they are linearly separable. The triplet network has clearly been trained successfully and the obtained representations generalize well to new images of the same materials. Furthermore, we see that the different colours, representing the different material groups, are also grouped together. This implies that the network has learned a similarity measure that closely mimics our notion of similarity, as we would also say that materials belonging to the same material group are more similar than materials belonging to different groups. Another impressive result is the fact that all this is feasible in only two dimensions. Apparently, it is possible to extract only two numbers from a microstructure image, so that these numbers already contain enough information to accurately tell the difference between 57 materials of which some are extremely similar. This result stresses the tremendous potential deep learning has in the analysis of microstructure images.

It is interesting to wonder how the quality of the representations changes when we include more than two dimensions. As was mentioned in the methodology, we measure the quality of the representations by computing the classification accuracy obtained by a random forest classifier. Intuitively, we would expect the accuracy to increase for increasing dimensionality as it becomes easier for the triplet network to spread out the representations in higher dimensional spaces. This intuition is confirmed in Figure 4.4, where we see that the accuracy monotonously increases with increasing dimensional-

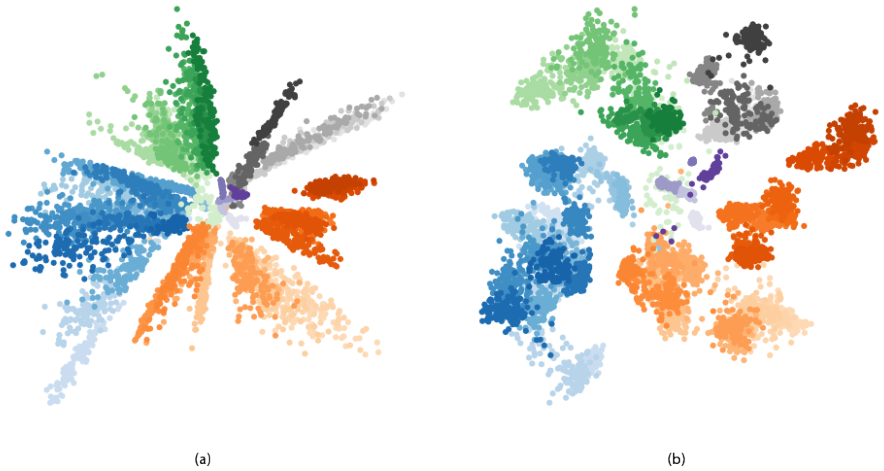


Figure 4.3: Microstructural representations in two dimensions obtained by (a) a CNN classifier and (b) the triplet network. Figure taken from [81].

ity of the representations. We have also included the classification accuracies for representations obtained from a CNN that was trained on the classification task by optimizing the cross-entropy loss. As for all other methods, the accuracies of the CNN classifier are obtained by training a random forest on the representations rather than directly using the classifications from the CNN. We do this to enable a fair comparison, as the CNN performs logistic regression on the representations, which is a linear classification method and might be inferior to random forests. We see that the triplet representations performs better than those of the CNN classifier for low dimensions. This indicates that the proposed method is very competitive with the state-of-the-art in microstructure recognition for images of materials on which the deep learning network was trained. The error bars are obtained by considering the three different training/test splits that were constructed as explained in Section 4.3. Overall, we see that the error bars are relatively small indicating that the model performance is not too dependent on the precise split of the training and the test set. For two dimensions, we obtain an accuracy of around 57 %, whereas for ten dimensions we can reach accuracies of about 68 %. Considering the benefits of having fewer dimensions, we conclude that the performance in two or three dimensions is already more than acceptable.

In Figure 4.5, we show the accuracies per magnification and per material group for the three-dimensional triplet model. It is clear that there is a clear preference for a specific magnification for each material group. For group one

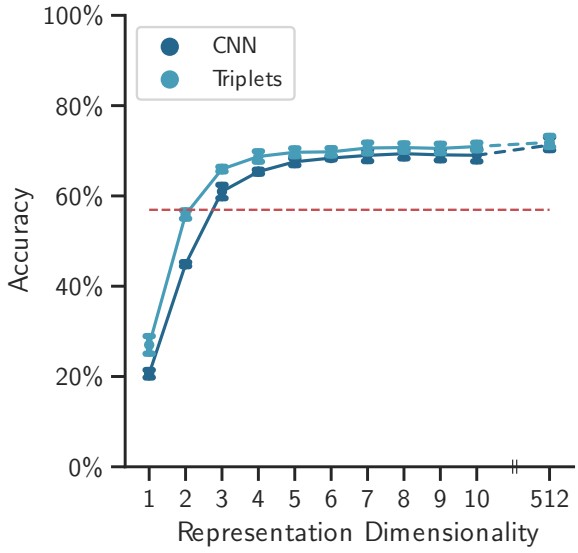


Figure 4.4: The classification accuracy as a function of the representation dimensionality for both the CNN classifier and the triplet network. Figure taken from [81].

and group two, an inter-pixel distance of 0.5 μm yields the highest accuracy. For group three, an inter-pixel distance of 0.1 μm is preferred. For group five, the inter-pixel distance should be larger than 0.1 μm in order to give optimal results. In general, we find that the model is able to cope well with the different magnifications in the dataset. The preference of the model for a specific magnification can be understood by considering that there is a trade-off between statistical representativity, which requires a sufficiently large area of the material to be on the image, and sufficient microstructural detail, which requires a sufficient level of magnification. From the results, it is clear that the optimal trade-off is strongly dependent on the material group under consideration.

The accuracies in Figure 4.5 suggest that materials in groups two, three and four are the hardest to correctly recognise. For groups two and three, this can be partially explained by the large number of materials in those groups. The more similar materials in the dataset, the harder it becomes to correctly distinguish between them. However, group four only contains nine different materials, which is not that much more than groups 1 and 5. The quenched martensites that belong to group four are visually all very similar and it might not be possible to tell apart the different materials purely based on single optical image crop. In the supplementary information of [81], we study

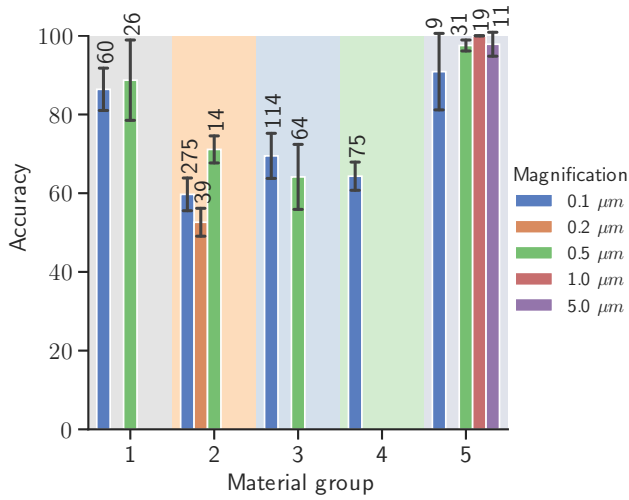


Figure 4.5: The classification accuracy per material group and per magnification, expressed as the inter-pixel distance, obtained using a random forest and the three-dimensional triplet representations. The numbers above the bars indicate the number of images belonging to that category. The error bars indicate the standard deviation. Figure adapted from [81].

more explicitly how the accuracies change when we decrease the number of material classes per subgroup.

Our results indicate that it is easily possible to obtain accuracies higher than 60 %. This is relatively high considering the visual similarity between the different materials. It triggers the question of where the model looks at. Such a question is important for two reasons. First, it helps us to better understand how the model works. As a deep neural network is essentially a black box containing millions of trainable parameters, every method that helps to elucidate the inner workings of the network is welcome. Second, it performs a sanity check of the results. In Ribeiro *et al.*[83] an example is given where a neural network is able to obtain a high classification accuracy for the task of telling a wolf apart from a husky. The deep learning model obtains a high accuracy, but upon analysis of where the model looks at, it turns out that the model mainly looks at whether there is snow in the background. If there is snow, the model automatically decides that the animal in the image is a wolf. While such a reasoning is not a priori bad, the model was not supposed to consider the background. It just turned out that all images of wolves had snow in the background. To avoid possible mistakes in the dataset

or in the model, analysing where the model looks at is definitely necessary. To shed light on which features the deep learning model deems important, we propose a simple method to obtain pixel-level predictions of the material class. We randomly sample 10 000 crops for a single image and classify each of these crops. The class predictions are assigned to each of the pixels that are in the crop. We can then compute per pixel a probability distribution of the pixel belonging to each of the classes. Such a pixel-level annotated image is also called a saliency map or a heatmap. In Figure 4.6, we show an heatmap of a microstructure image. From the heatmap it is clear that the model mainly looks at the density of the pearlitic lamellae to determine to which material the image belongs. The blue regions on the heatmap contain notably less lamellae and are assigned to the "confused material class", which also exhibits notably lower densities in lamellae. In other examples that are discussed in the reference paper[81], we find that also features such as the size and the shape of grains are considered important by the model. On the other hand, we also find that the model can be deceived by etching artefacts. This seems understandable as the model only looks at the data and is not able to make a distinction between physical features of the microstructure and artefacts due to processing or etching. Still, we can conclude that the model looks at physically relevant features in the microstructure to compute the representations. More heatmaps and analyses of examples where the model fails can be found in the supplementary information of the reference paper.[81]

It is interesting to compare the classification accuracy for our highly compact representations of the microstructure to that of other microstuctural representations commonly used in the literature. In Table 4.3, we list the performances of several commonly used representations. The Haralick[45] and texture[46] features are both derived from the textural features proposed by Haralick, as we discussed in the previous chapter. The two-point static features were also discussed before. The VGG16 mean C_{43} features are extracted from the penultimate convolutional block of a VGG16 network that was pretrained on ImageNet[68] and averaged out in the spatial dimensions. As before, the errors are obtained by using three different splits for the training and the test set. We see that the two-dimensional triplet representations already achieve an accuracy that is on par with the best methods in the literature. The three- and ten-dimensional representations outperform the other methods by a quite a large margin, despite being lower in dimensionality. A comparison to a larger set of microstructural representations found in the literature is included in the supplementary information of the reference paper.[81]

To assess the performance of the representations on new materials, we

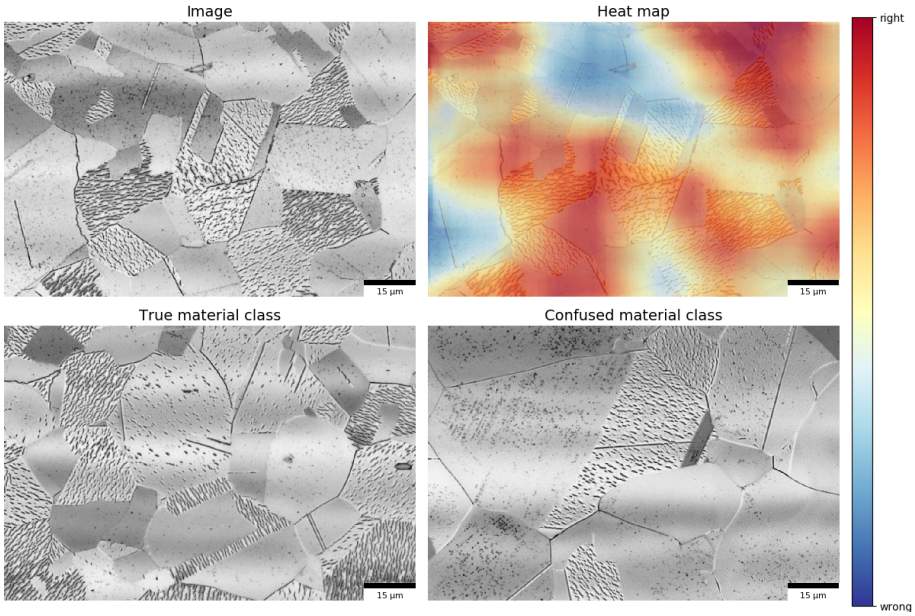


Figure 4.6: An example of a saliency map for the predictions of the three-dimensional triplet network. Figure adapted from [81].

method	dimensionality	test acc. group[%]	test acc. mat.[%]
Haralick[45]	13	94.4 ± 0.2	46.5 ± 2.3
texture[46]	8	90.6 ± 1.5	44.4 ± 3.8
two-point statistic[49]	20	78.0 ± 1.0	19.7 ± 1.9
VGG16 mean C_{43}[75]	512	99.2 ± 0.3	56.2 ± 1.5
CNN classifier	2	98.3 ± 2.0	44.8 ± 0.6
CNN classifier	3	99.0 ± 2.1	60.5 ± 1.7
CNN classifier	10	98.9 ± 1.6	69.3 ± 1.7
Triplets	2	98.8 ± 1.8	55.6 ± 1.1
Triplets	3	99.1 ± 1.8	65.9 ± 0.7
Triplets	10	99.2 ± 1.5	71.0 ± 1.2

Table 4.3: Comparison of the classification accuracy of several microstructural representations on dataset 1. We give the accuracies (acc.) on the test set obtained by a random forest classifier both for the different groups and for the different materials. Table adapted from [81].

evaluate the different representations on dataset 2. The results are listed in Table 4.4. We show the out-of-bag accuracy, which is computed based on the training data, and the test accuracy. In all cases, the out-of-bag accuracy is significantly higher than the test accuracy. This is because the crops used in the training set can overlap, making it easier for the model to correctly recognise the material. It stresses the importance of using different images in the training and test set. Unlike for the first dataset, the triplet representations only score slightly better than other representations of the same dimensionality. This suggests that the model does not generalize as well to new materials as we would like. It was to be expected, as for instance the neural network has never seen bainitic microstructures during the training. Therefore, it is not able to recognise the most important microstructural features in such structures. The generalization can most likely be mitigated by using an even bigger and more diverse dataset to train the representations on. A second thing to note, is the strong correlation between the dimensionality and the accuracy. It is clear that higher dimensional representations generalize better to new microstructures. A last important observation is that the representations obtained from a CNN classifier perform better at generalizing towards new datasets. This finding is in line with the literature. For instance in the recent work done by Khosla *et al.*[84], it is reported that networks trained using the cross-entropy are competitive with more advanced versions of the triplet networks when it comes to transferability to new datasets. A comparison to a larger set of microstructural representations found in the literature is included in the supplementary information of the reference paper.[81]

method	dimensionality	out-of-bag acc.[%]	test acc.[%]
Haralick[45]	13	98.4 ± 0.3	91.5 ± 3.2
texture[46]	8	89.6 ± 1.9	83.4 ± 2.2
two-point statistic[49]	20	79.8 ± 1.0	69.8 ± 2.6
VGG16 mean C_{43}[75]	512	99.4 ± 0.1	97.8 ± 0.5
CNN classifier	2	71.3 ± 5.3	67.1 ± 5.0
CNN classifier	3	84.4 ± 1.6	79.1 ± 2.6
CNN classifier	10	97.9 ± 0.8	95.9 ± 1.8
Triplets	2	71.7 ± 4.8	66.2 ± 6.5
Triplets	3	83.7 ± 3.9	78.9 ± 5.3
Triplets	10	97.7 ± 0.6	94.6 ± 2.0

Table 4.4: Comparison of the classification accuracy of several microstructural representations on dataset 2. Both the out-of-bag accuracy (acc.) and the accuracy for the test data are listed. Table adapted from [81].

4.5 Conclusion

In this chapter, we discussed triplet network as a means to learn low-dimensional representations of microstructure images. Already in two dimensions, the representations contain enough information to distinguish between visually very similar materials. Although higher-dimensional representations can contain even more information, we find that two or three dimensions are enough to describe a microstructure in acceptable detail. The low dimensionality of the representations is promising, as it allows us to build robust machine learning models that take microstructural information as input. By analysing saliency maps, we find that the triplet network automatically identifies physically relevant features in the image. For the set of materials on which the deep learning model was trained, we find that the triplet representations are better able to discern the different materials than other microstructural representations found in the literature. However, this is no longer the case for new materials where the discriminative power of the triplet representations is only on par with the other methods found in the literature and is slightly inferior to the performance of a CNN classifier that is trained using the cross-entropy loss. This suggests that the presented method is mainly useful for applications where a set of predefined materials is analysed. We discuss such an application in the next chapter.

One of the main advantages of the presented method to learn representations, is the fact that the representations will become better and more general

if larger datasets of images are used. The dataset used in this work contained a total of 780 images. If larger and more diverse datasets become available in the future, the presented method will further gain in relevance. In the ideal case, a dataset such as ImageNet could be constructed purely consisting of microstructure images of steel. The availability of such a dataset could revolutionize the way in which microscopy images are analysed.

5

Microstructure recognition using deep learning

There is no reason and no way that a human mind can keep up with an artificial intelligence machine by 2035.

Gray Scott, futurist

5.1 Introduction

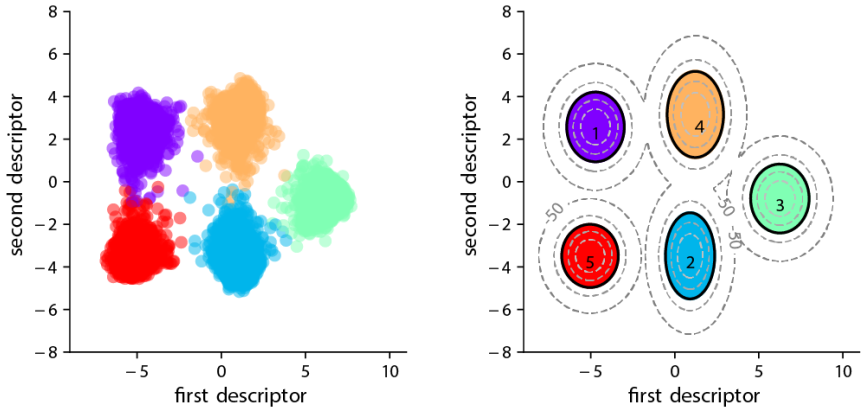
In the previous chapter we demonstrated how triplet networks can be used to obtain microstructural representations that can discern between visually very similar microstructures. We suggested that these structures are so similar that even an expert metallurgist would struggle to see the difference. This raises an interesting question: how well does deep learning perform compared to an expert metallurgist in recognising microstructures? We mentioned before that Ciresan *et al.*[53] already achieved superhuman performance in traffic sign recognition back in 2012. Since then, deep learning has seen tremendous improvements both in algorithms and in hardware, so that it would be interesting to compare the performance of deep learning and human experts on the same microstructure recognition task. Ciresan *et al.* competed with non-expert people, whereas we are dealing with experts for whom the pride in their expertise is at stake. Reports about deep learning models outperforming domain experts have already been published

in medicine.[85, 86] To verify whether this can also be achieved in the microstructure analysis of steel, we organize two quizzes in which both experts and non-experts compete against a deep learning model to correctly recognise microscopy images of steel. An additional challenge for the deep learning algorithm is that the amount of training data is limited. This requires an extremely data-efficient approach to train the network.

5.2 Methodology

We use a two-step approach. In the first step, we train a triplet network to obtain detailed and low-dimensional representations of the microstructure images. In the second step, we train a machine learning model to assign each of the representations to a material class.

The training of the triplet network is performed in a similar fashion to what we discussed in the previous chapter, although there are a few important differences in order to make the training more data-efficient. First, we use the ResNet50 architecture[70] rather than the ResNeXt50 architecture[71], as we found that the former is easier to optimize when dealing with smaller datasets. We also use the Adam optimizer[64] instead of SGD as this leads to faster convergence of the triplet network. The most important difference is that we implemented some new image transformations to artificially augment the number of images in the dataset. This is called image augmentation and is commonly used to train models so that they generalize better to new data. In addition to standard image transformations such as rotating, mirroring and changing the brightness of the image, we implemented a set of transformations that are specifically relevant in the context of microstructure analysis. The first transformation is the local blurring of the microstructures using Gaussian filters. Such a transformation mimics the presence of regions of the image that are out of focus due to for instance the presence of a dust particle on the sample. A second transformation, is the inclusion of edge detectors to better highlight the grain edges in the images. This helps the model to understand the importance of the grains. Lastly, we warp the image around its boundaries to stress the translational invariance in conjugation with the other transformations. Unlike in the previous chapter, we do not generate the crops before the training, but we randomly select crops from the original images during the training of the model. This has the advantage that there is more variety in crops as the exact same crop will never be shown twice to the network. Furthermore, we vary the size of the crops. Rather than only selecting crops of a 200×200 pixel resolution, we select crops that can be slightly smaller or bigger. These crops are then interpolated,



(a) triplet network **(b)** Gaussian mixture

Figure 5.1: The output of the respective steps in the methodology. The triplet network (a) yields microstructural representations. The Gaussian mixture model (b) differentiates between in- and outliers. The dashed grey lines show the log-likelihood.

so that they end up with a pixel resolution of 200×200. This approach assures that information from different length scales is included. A last important modification is that we also explicitly include information about the magnification. This is useful to help the network deal with the large differences in magnifications between the different images.

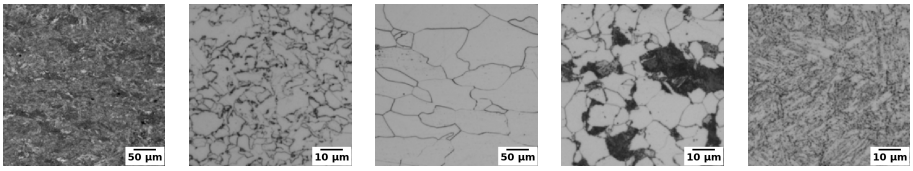
In the previous chapter, we used a random forest classifier to assign the microstructural representations to a material. In the quizzes, however, there is a possibility that the shown image does not belong to any of the predefined materials. To account for this possibility, we use a Gaussian mixture model.[87] This type of model fits a multivariate Gaussian distribution to each of the material classes. The parameters of the distribution are optimized by maximizing the likelihood of the training data. Because we fit a distribution to the data, it is possible to compute the likelihood of a new data point. If the likelihood of data point falls under a certain threshold, we can consider this point as an outlier. We select the threshold so that 99.7% of the training data are inliers. This number is inspired on the three-sigma rule for normal distributions.[88] The approach is illustrated in Figure 5.1. In the left figure we show the two-dimensional triplet representations which serve as the input for the Gaussian mixture model. In the right figure, we show the results of the model. If a representation lies in one of the coloured regions, it will be assigned to the respective class. However, if it falls outside of those

regions, it will be considered an outlier.

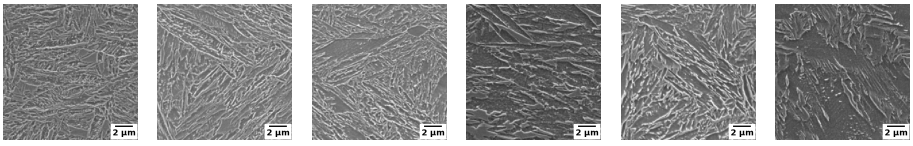
Two quizzes are organised. Each quiz consists of a set of images that has to be assigned to one of the pre-defined classes. The participants received a link through which they could take the quiz. There was no time limit for them. The participants were divided into two categories: experts and non-experts. In order to be considered an expert, one had to have a PhD in materials science and/or several years of working experience in the field of metallography. The participants were all shown a few example images of every class. They did not see the entire dataset on which the deep learning model was trained. To evaluate the quiz images with the deep learning model, we randomly select 1 000 crops from each image and compute the predictions with the workflow described above. We average the predictions of the 1 000 crops to obtain the predicted material class for the image. We use this approach to obtain more meaningful predictions. There is always the possibility that not all relevant information is included in a single 200×200 crop. By averaging over many different crops, we aim to include all information present in the image.

5.3 Datasets

We organize two quizzes each with a different dataset. The first quiz covers five different types of microstructures and contains both OM and SEM images. The OM and SEM images do not necessarily depict the same samples. A description of each class is given in Table 5.1. In this quiz, the "none of these" option is included both for the participants and for the deep learning model.



(a) quiz 1



(b) quiz 2

Figure 5.2: Some examples of the crops that are used to train the deep learning models for both quiz 1 (a) and quiz 2 (b). Figure taken from [79].

For each class, there are 15 to 52 images in the dataset with magnifications ranging from 1.1 pixels per micrometer to 212 pixels per micrometer. To address the unbalance in the number of images per class, we sample more crops from the images belonging to the under-represented classes. The quiz itself contains 36 images. The number of quiz images per class is also shown in Table 5.1.

class	description	appearance	# images
1	tempered martensitic	needle-like structure and smoother zones	52 - 6
2	bainitic	quasi-polygonal grains with rough boundaries	24 - 7
3	ferritic	very coarse polygonal grains	17 - 5
4	ferritic/pearlitic	quasi-polygonal grains and platelets	15 - 5
5	quenched martensitic	sharp needle-like structure without carbides	26 - 4
6	none of these	what doesn't belong to the other classes	0 - 9

Table 5.1: A description of the different material classes included in quiz 1 and respectively the number of images used for training the model and the number of images included in the quiz for each class. Class six has by definition no images that are included in the training set. Table adapted from [79].

For the second quiz, we consider six different types of complex martensitic structures. These are described in Table 5.2. In this quiz, there is no "none of these" option for the participants, but we have included it for the deep learning model. For each material class there are only four to twelve images with magnifications ranging from 53 to 106 pixels per micrometer. Again, we sample the crops in such a way that the network sees the same number of crops for each class. The quiz contains 21 images. The number of quiz images per class is also shown in Table 5.2. More information on the datasets that are used in this chapter can be found in the supplementary information of the reference paper.[79]

class	appearance	# images
1	austenite in the shape of relatively isolated equiaxed blocks	12 - 4
2	lath-like austenite and martensite with visible carbides	12 - 3
3	sharp needle-like martensitic structure	4 - 3
4	coarse, blocky austenite areas	12 - 4
5	austenite in the shape of blocks and laths without visible carbides	6 - 3
6	highly oriented lath structure with coarse carbides	12 - 4

Table 5.2: A description of the different material classes included in quiz 2 and respectively the number of images used for training the model and the number of images included in the quiz for each class. Table adapted from [79].

5.4 Results

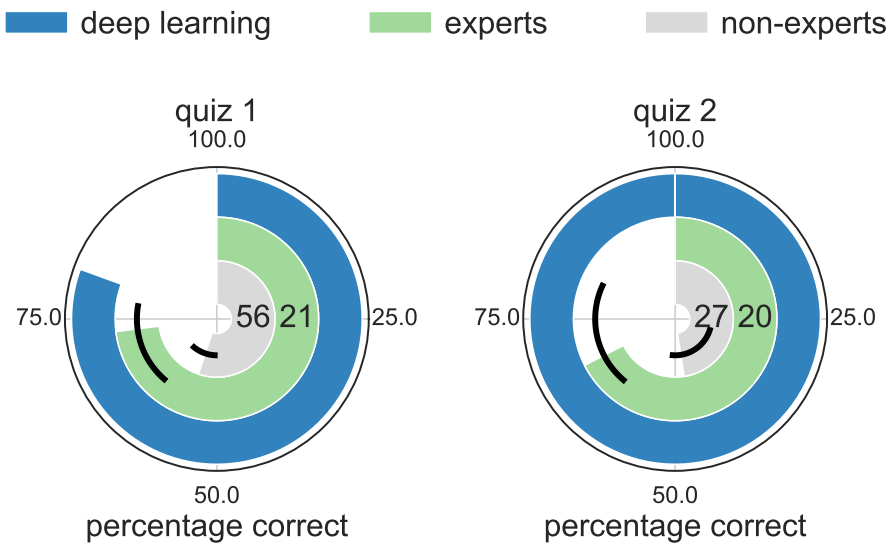
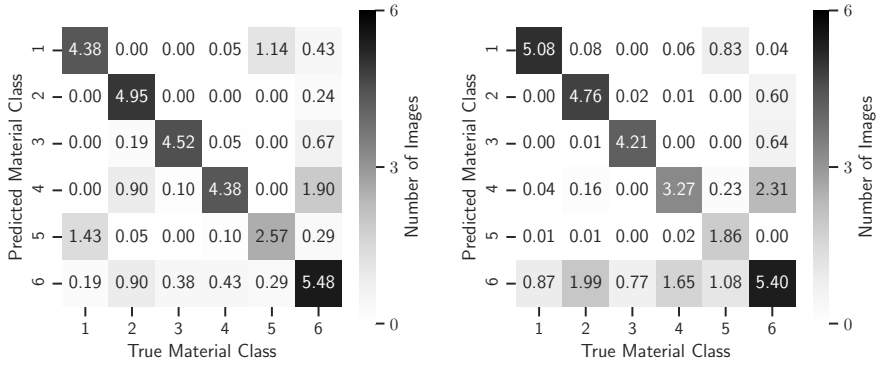


Figure 5.3: The results of both quizzes. The numbers in the bars show the number of participants for the corresponding category. The error bars indicate the interquartile range for the experts and non-experts. Figure adapted from [79].

In Figure 5.3, we show the percentage of correctly answered questions of the quiz for the non-experts, experts and deep learning model. We see that in both cases the deep learning model obtains the highest performance. As expected, the experts outperform the non-experts by quite a large margin. It is clear that the non-experts struggle the most with the second quiz, where they answer on average less than 50 % of the questions correctly. This is unsurprising, since distinguishing between different types of complex martensitic microstructures requires some experience. On the other hand, it is notable that the experts obtain a similar score for both quizzes, while the second quiz could be considered much harder. The reason for the similar scores is most likely the presence of the "none of these" option in the first quiz, which seems to cause a lot of confusion. If we look at the scatter on the scores of the experts, we see that there is a slightly larger spread on the scores for the second quiz. This can be explained by noting that not all experts were equally familiar with the specific complex martensitic steels that were examined in the second quiz.

As is shown in the confusion matrix in Figure 5.4, the option of the material being an outlier causes a lot of confusion for the deep learning algorithm. In



(a) experts **(b)** deep learning

Figure 5.4: The confusion matrices for the first quiz for both the experts and the deep learning model. For the experts, we give the average confusion matrix per expert, whereas for the deep learning model we use the probabilities rather than the predictions.

six of the in total seven misclassified images by the model, this plays a crucial role. While we established an easy and intuitive criterion for detecting outliers, it is clear that the determination of outliers remains somewhat arbitrary. If we do not consider the last row and column, we see that the deep learning algorithm obtains a very good performance with very low values for the off-diagonal elements. Only in differentiating between class 4 (ferrite/pearlite) and class 5 (quenched martensite), the deep learning model performs worse than the averaged predictions of the experts. A more detailed overview of the answers and predictions per image can be found in the supplementary information of the reference paper.[79]

An instructive example where the model incorrectly fails to recognise the material is shown in Figure 5.5. The material on the image is pearlitic and is shown at low magnification. For trained metallurgists, it is easy to recognise the lamellar structure even when they are typically studied at higher magnifications. The deep learning model, that has only seen images of pearlite at a magnification that is at least 2.5 times higher, does not recognise the lamellar structure and incorrectly considers this image to be an outlier. While we included some scale invariance by selecting crops at different sizes, it is impossible to overcome a magnification factors of 2.5 with this approach. The only way to prevent the model from making this mistake is by including such examples in the training set.

One could argue that it was mistake not to include more relevant examples

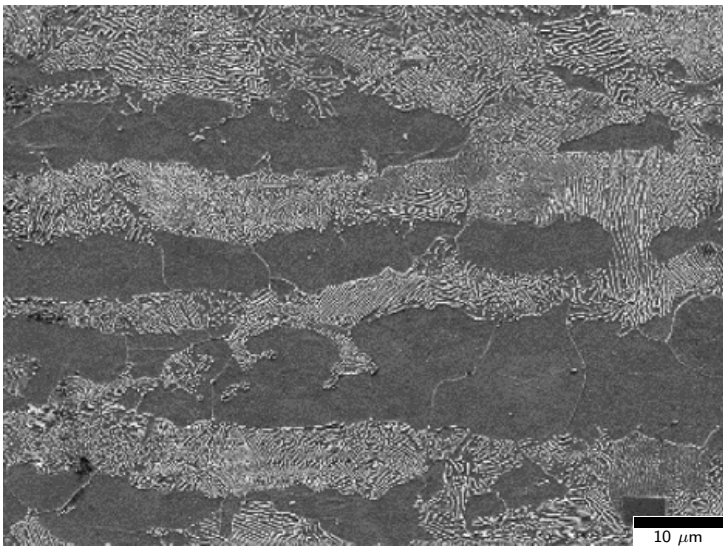
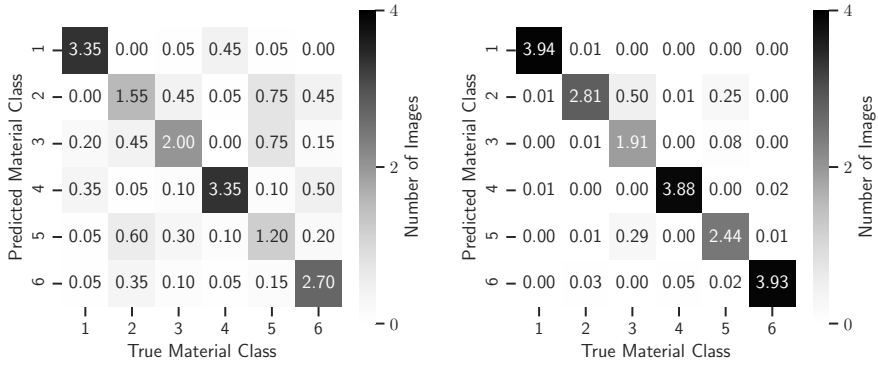


Figure 5.5: One of the few quiz images that was misclassified by the deep learning model for the first quiz. Figure adapted from [79].

in the training set, so that the model could perform even better at the quiz. However, one of the purposes of the first quiz was to better understand to which extent a deep learning model is able to withstand changes in etching, image quality, illumination and magnification. The images in the quiz were selected by experts to be challenging for a machine learning algorithm. It is only thanks to the extremely aggressive data augmentation that the model is still able to obtain a very good score that could compete with the best experts. As in the previous chapter, we find that mainly the etching procedure is able to deceive the deep learning model. This is discussed in more detail in the supplementary information of the paper, where we also show some heatmaps of the model predictions for the first quiz.

For the second quiz, we see that the deep learning model obtains a perfect score. This might seem to be suspiciously good, but among the experts there were also a few people who managed to obtain a close to perfect score. From the confusions matrices shown in Figure 5.6, it is clear that the off-diagonal elements of the confusion matrix of the deep learning model are typically smaller than those of the confusion matrix of the experts. The only exception is the confusion between class two and class three. We conjecture that this is due to the presence of precipitates in some regions of the images of class two which confuses the model, as we will discuss in the next paragraph. Compared to the first quiz, it is remarkable how confident the model is in its predictions. We see that the model is the most unsure for images belonging



(a) experts **(b)** deep learning

Figure 5.6: The confusion matrices for the second quiz for both the experts and the deep learning model. For the experts, we give the average confusion matrix per expert, whereas for the deep learning model we use the probabilities rather than the predictions.

to classes three and five, which are the classes that have the fewest images in the training set. It is plausible that in order to make confident predictions the model needs to have seen as many examples of the material class as possible. Still, it remains impressive that the model is able to make confident and correct predictions when having seen only up to twelve images per material class. A more detailed overview of the answers and predictions per image can be found in the supplementary information of the reference paper.[79]

One of the most interesting examples of where the models doubts is shown in Figure 5.7. The model mainly doubts between classes two and three. To shed more light as to why this is the case, we have also included a heatmap that is obtained using the same procedure that was described in the previous chapter. From the heat map, it is clear that mainly the lower corners of the images are assigned to class two, as those regions are marked in blue. A closer inspection of these regions shows that they contain some fine precipitates. As class two is the only class with such fine carbides, it is understandable for the model to assign these regions to that class. The rest of the image is correctly assigned to class three.

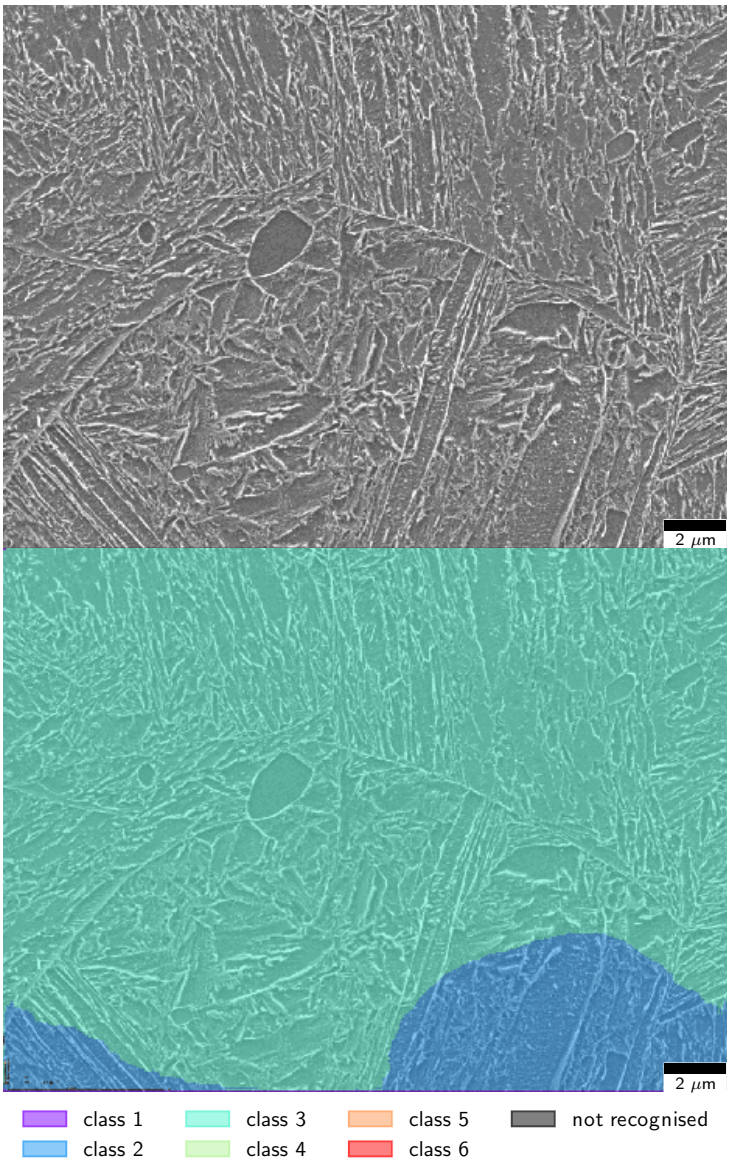


Figure 5.7: One of the examples where the deep learning model was not very confident in its prediction for the second quiz. Figure adapted from [79].

5.5 Conclusion

In this chapter, we let a deep learning model compete with human experts in a microstructure recognition problem. We presented a methodology that is highly data-efficient and that is able to deal with outliers in the dataset. By including image transformations that are tailored specifically to the microscopy imaging of steels, we can train models that generalize well to new data. The obtained deep learning models are able to compete with the best experts. The most surprising result, however, is that this is possible with only a small number of images per material class. Deep learning models seem to be especially suited when dealing with complex martensitic steels of which the microstructure contains many small features. When looking at where the models fail, the results indicate that dealing with outliers remains difficult as a somewhat arbitrary choice has to be made of when a data point can be considered to an outlier. As before, we find that the etching procedure can deceive the model. Upon inspection of the heatmaps, we can conclude once more that the model looks at physically relevant features in the microstructure.

The presented methodology can be especially useful in industrial settings, where an automated inspection of the quality of the steel is required. The possibility of detecting outliers that need further inspection, makes this method extremely suited for such tasks. The low-dimensional representations of the microstructure in combination with the heatmaps could enable operators to better understand why the model is making certain decisions.

6

Structure-property prediction

6.1 Introduction

One of the main aims of this PhD is to investigate how deep learning can be used to shed light on the links between the processing, the structure and the properties, previously referred to as the PSP-linkages. This is essential for the design of new materials if one wants to avoid trial-and-error approaches, as is illustrated in Figure 6.1.[89] However, establishing such linkages is not an easy task. Unlike microstructure recognition, where we deal with a predefined set of materials, the obtained models need to generalize to new materials. Furthermore, there is no a priori guarantee that a single microscopy image contains enough information about the material to accurately determine its composition, processing or its properties. In recent work, Yucel *et al.*[90] have used the two-point statistic to link optical microscopy images of the microstructure to the toughness of steel. From chapter 4, we know that deep learning offers possibilities to describe the microstructure in more detail than the two-point statistic. Furthermore, it has already been shown in the literature that deep learning models can learn the relation between the microstructure and the properties for data that has been simulated by finite element models.[91, 92] It is therefore safe to state that deep learning holds great promises in building PSP-links based on experimental data.

As such experimental data is scarce, we investigate two simple, data-efficient approaches to obtain robust PSP-links. We demonstrate the practical value

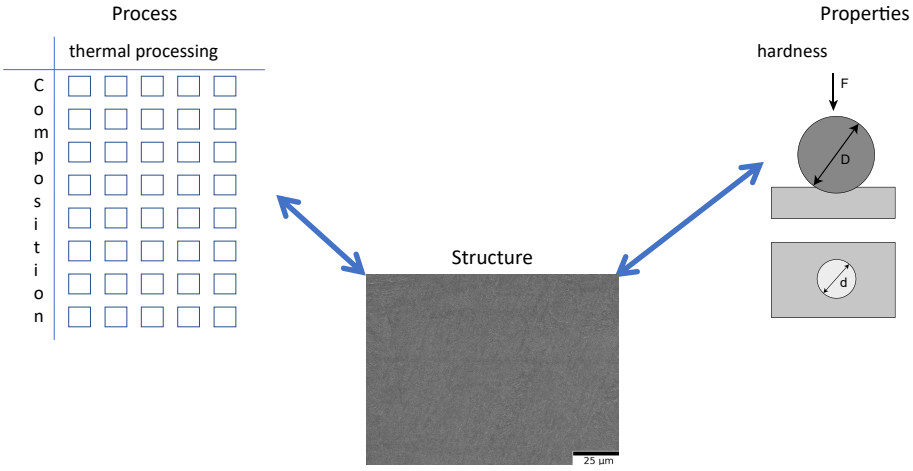


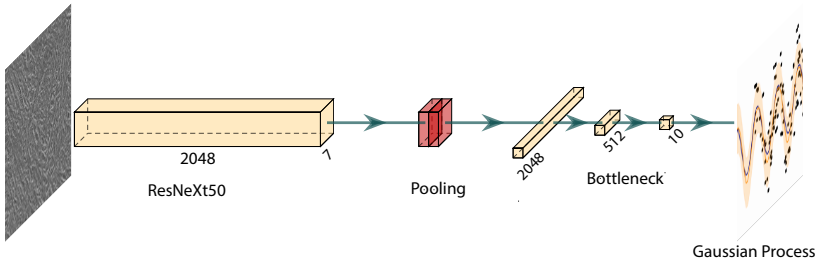
Figure 6.1: Accurate PSP-links would allow metallurgists to explore different combinations of composition and processing without having to make the steel in order to assess the resulting microstructures and properties.

of our approaches by establishing links both between the structure and the composition and between the structure and the hardness for a set of complex martensitic steels. We assess the performance of our structure-property model by comparing it to a model that uses compositional information to predict the hardness.

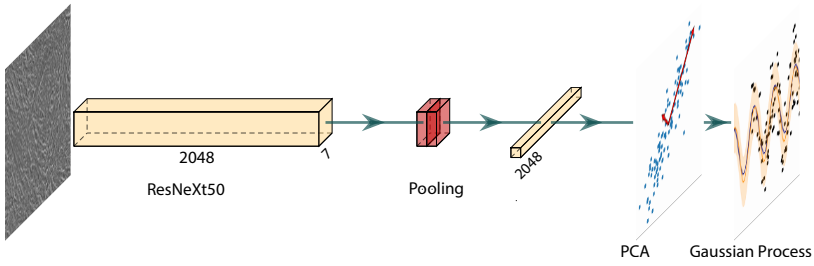
6.2 Methodology

We test two different approaches to establish PSP-links, as is illustrated in Figure 6.2. For the first approach, which is shown in Figure 6.2a, we compute ten-dimensional microstructural representations using a CNN classifier network that was trained on a dataset of similar microstructures. We choose the ten-dimensional CNN network, as we found in chapter 4 that higher-dimensional representations of CNNs tend to generalize better to new datasets. As input for the network, we do not use 200×200 crops as we did before, but we use the entire image at once to obtain a single representation per image. The microstructure representations serve as input for a Gaussian process regression model.[93]

In chapter 4, we noticed that the performance of deep learning representations becomes worse for new, unseen materials. Furthermore, we are here considering a regression problem rather than a classification problem as we



(a) CNN classifier representations



(b) PCA features

Figure 6.2: An illustration of the two methods studied to establish PSP links.

did before. This implies that the representations should not only generalize to new materials, but also to new tasks. Given the results in chapter 4, this is highly unlikely. We therefore consider a second approach, shown in Figure 6.2b, where we extract intermediate output from the CNN network right after the adaptive average pooling layer. From DeCost *et al.*[75], it is clear that output from intermediate layers tends to generalize better to new materials and new tasks. The vector representations that come out of the adaptive average pooling layer have a size of 2048, which is too high for our purposes. As most regression methods work best in low-dimensional spaces, we use PCA to reduce the dimensionality of the extracted features. We keep the number of retained principal components fixed to ten, which retains more than 60 % of the total variance in the data. These principal components serve as input for a Gaussian process regression model. As the presented approaches both require two different models to be trained, the CNN classifier and the Gaussian process, we repeat all experiments three times to get an idea of the spread on the model performance.

A Gaussian process is a regression model that assumes the data to be jointly normally distributed. The prediction f_* for a new material is then also normally distributed. Given the input features $\boldsymbol{x}_*$ of a new material, the

training inputs X and the training targets $\boldsymbol{y}$, we have[93]

$$f_* \sim \mathcal{N}(mean(f_*), var(f_*))$$

with

$$mean(f_*) = K(\boldsymbol{x_*}, X)\left[K(X,X) + \sigma I\right]^{-1}\boldsymbol{y}$$

and

$$var(f_*) = K(\boldsymbol{x_*}, \boldsymbol{x_*}) - K(\boldsymbol{x_*}, X)\left[K(X,X) + \sigma I\right]^{-1}K(X, \boldsymbol{x_*}).$$

The kernel $K(\boldsymbol{x}, \boldsymbol{x}^{'})$ can be interpreted as a similarity measure between the different data points and σ represents the noise level on the data. The most popular choice for the kernel is the radial basis function kernel

$$K(\boldsymbol{x}, \boldsymbol{x}^{'}) = \exp\left(-\sum_{i=1}^{N}\frac{(\boldsymbol{x}_i - \boldsymbol{x}_i^{'})^2}{\boldsymbol{l}_i}\right),$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_i^{'}$ are the i-th components of the N-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$ respectively. In this chapter we only use this type of kernel. The free parameters σ and $\boldsymbol{l}_i$ are optimized by maximizing the likelihood of the training data, so that no additional validation set is necessary. The variance $var(f_*)$ on the predicted value expresses how confident the model is in its predictions. The fact that Gaussian processes are able to provide us with such a reliable uncertainty estimate and the fact that a validation set is not needed are the two main reasons why we prefer to work with Gaussian processes in this chapter.

Despite not needing a validation set, we still need to split the dataset into a training and a test set. As we only consider 52 materials in total, it is not easy to construct a training and test set that are fully representative for all materials in the dataset. We therefore resort to leave-one-out cross-validation[94] to obtain a good assessment of the model performance. We omit one material and train the Gaussian process on the remaining 51 materials. A prediction is then made for the omitted material. This procedure is repeated for all materials, so that we obtain predictions for all materials while they are not included in the training set. Such a Leave-one-out cross-validation procedure gives a good idea of the generalization to new, unseen materials.[78] The downside of this approach is that it requires 52 different Gaussian process models to be trained.

A last thing we need to decide on, is which metrics we want to use to evaluate the model performance. A good metric has to meet two requirements. First, it should be easy to interpret, so that metallurgists, who are usually not

acquainted with machine learning, can get a good idea of the model performance. Second, it should allow us to compare the predictive performance between different properties. As we found no metric that fully satisfies both requirements at the same time, we use two different metrics. The first metric is the mean absolute error (MAE), which is defined as

$$MAE(\boldsymbol{y}, mean(\boldsymbol{f})) = \frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{y}_i - mean(\boldsymbol{f}_i)|,$$

where $\boldsymbol{y}_i$ and $mean(\boldsymbol{f}_i)$ are respectively the experimental value and the predicted mean value for sample i and N is the number of samples in the dataset. This metric has the advantage of being simple to interpret: the MAE gives the expected absolute difference between the model prediction and the experimental value. The second metric we use is the R^2-score, defined as

$$R^2(\boldsymbol{y}, mean(\boldsymbol{f})) = 1 - \frac{\sum_{i=1}^{N}(\boldsymbol{y}_i - mean(\boldsymbol{f}_i))^2}{\sum_{i=1}^{N}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^2},$$

where $\bar{\boldsymbol{y}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{y}_i$ is the average value of $\boldsymbol{y}$. The second term in the definition can be interpreted as the ratio of the mean squared error, which is defined in equation (3.4), to the (biased) variance on the experimental values. Higher values of the R^2-score indicate a lower MSE and therefore a better fit. For a perfect fit, the MSE is zero and the R^2-score equals one. The advantage of the R^2-score is that it is scale-independent, which allows us to compare the model performance for different properties.

6.3 Datasets

Two datasets are used in this chapter. The first dataset contains FEG SEM images of 26 complex martensitic steels that have differences both in composition and thermal processing. For each steel there are 34 images with magnifications ranging from $\times 250$ to $\times 20\ 000$. Each image has a pixel resolution of 1280×960. This dataset is used to train the CNN classifier network with each material representing a class.

The second dataset contains 52 complex martensitic materials that differ both in composition and thermal processing. For each material, 10 FEG SEM images were taken at 4 different magnifications: $\times 1\ 000$, $\times 2\ 500$, $\times 5\ 000$ and $\times 10\ 000$. Each image has a pixel resolution of 1280×960. Some example images are shown in Figure 6.3. For all materials, the Brinell hardness (HBW 2.5/187.5) was measured.[95] This dataset is used to establish and evaluate the PSP links.

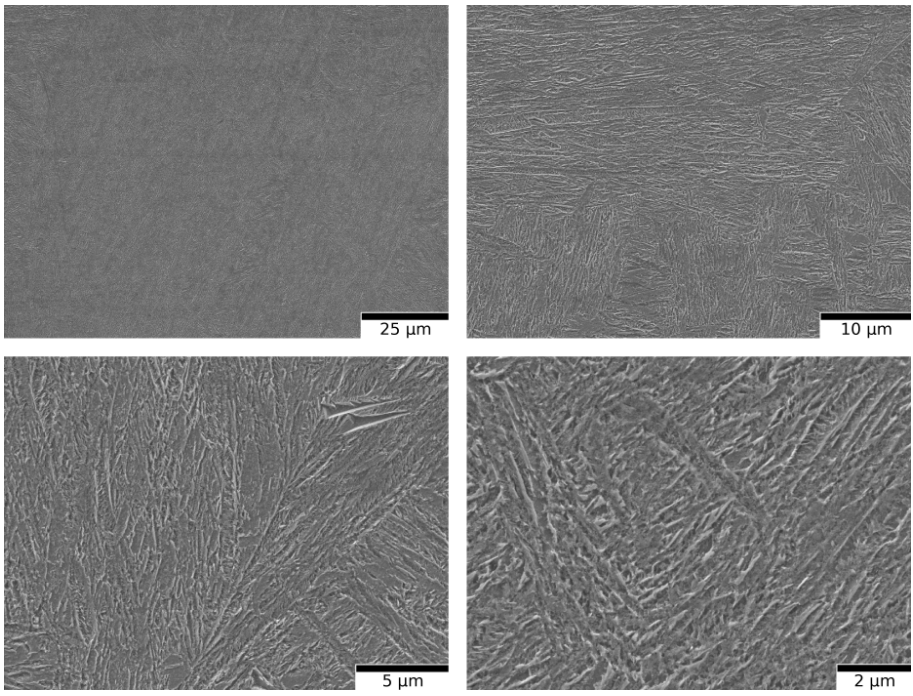


Figure 6.3: Some examples of the SEM images that are used in this work. We consider four different magnifications of 52 complex martensitic steels. The central question is how much these images reveal about the hardness and the composition of the steel.

6.4 Results

In this section, we study two types of links: the structure-hardness link and the structure-composition link. The former is relevant to avoid complex and possibly destructive measurements when this is not useful based on the model prediction. The latter helps to elucidate how much information about the composition of the steel is included in the microstructural representations. In order to build accurate structure-property models it is necessary that the information about the composition is somehow encoded in the microstructural representations. By comparing the performance of the structure-hardness model to that of a model that only uses information about the composition to predict the hardness, we can assess how much additional information about the material is included in the microstructural representations.

In Table 6.1, we show the R^2-scores for the predictions of the hardness and the composition based on a single image. The CNN representations clearly

Property	Magnification	R^2-score CNN	R^2-score PCA
hardness	x1 000	0.50 ± 0.14	0.64 ± 0.03
hardness	x2 500	0.42 ± 0.11	0.54 ± 0.01
hardness	x5 000	0.40 ± 0.12	0.55 ± 0.07
hardness	x10 000	0.31 ± 0.07	0.45 ± 0.02
carbon content	x1 000	0.57 ± 0.07	0.83 ± 0.04
carbon content	x2 500	0.62 ± 0.02	0.88 ± 0.02
carbon content	x5 000	0.66 ± 0.07	0.80 ± 0.08
carbon content	x10 000	0.59 ± 0.11	0.76 ± 0.06
manganese content	x1 000	0.31 ± 0.06	0.36 ± 0.19
manganese content	x2 500	0.12 ± 0.04	0.56 ± 0.04
manganese content	x5 000	0.30 ± 0.12	0.37 ± 0.11
manganese content	x10 000	0.31 ± 0.08	0.34 ± 0.04
silicon content	x1 000	0.15 ± 0.03	0.14 ± 0.02
silicon content	x2 500	0.06 ± 0.08	0.15 ± 0.08
silicon content	x5 000	0.01 ± 0.06	0.11 ± 0.07
silicon content	x10 000	0.12 ± 0.06	0.07 ± 0.08

Table 6.1: The R^2-score per magnification for the hardness and composition based on a single SEM image. We show the scores both for the ten-dimensional CNN representations and for the PCA features extracted from the adaptive average pooling layer. The error estimates are the standard deviations computed based on three different CNNs.

yield an inferior performance to the PCA features, which seem to generalize better. We will therefore limit our discussion to these features. The R^2-score for the carbon content is around 0.8 for moderate magnifications. This means that about 80 % of the variance in the dataset can be explained by the model and suggests that it is possible to accurately estimate the carbon content based on a single SEM image. For the hardness, we obtain R^2-scores well above 0.5 for lower magnifications. This is still relatively good, which was to be expected as the carbon content and the hardness of steel are typically strongly correlated.[96] For predicting the manganese content, we see that there is a clear preference for the $\times 2\ 500$ magnification for which we also obtain a R^2-score above 0.5. This is not the case for silicon, where the R^2-scores are relatively close to zero. This suggests that it is not possible to determine the silicon content from SEM images. Overall, we find that there

is a preference for lower magnifications. This can be explained by the fact that lower magnifications give a larger and thus more representative view of the material under consideration. If we look at the scatter among the three different runs, we see that the spread on the performance of the PCA features is acceptable.

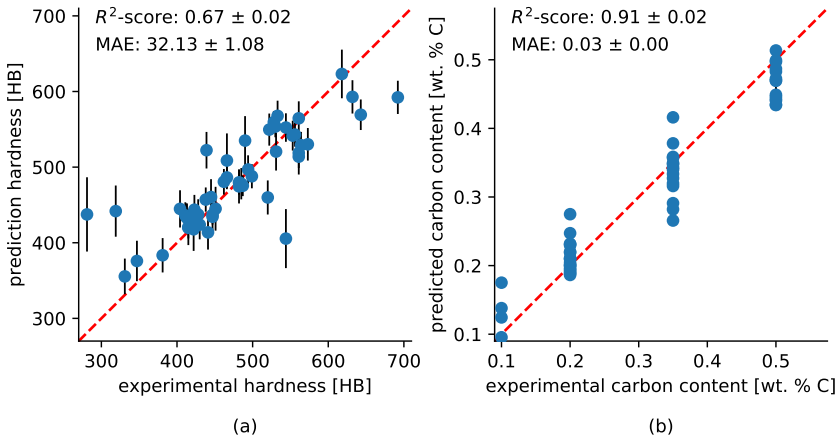


Figure 6.4: A scatter plot of the predicted values against the targeted values for both the hardness (a) and the carbon content (b) using the average of the predictions of all 40 images per material. All predictions are obtained using leave-one-out cross-validation. For a perfect model, all points would lie on the diagonal indicated by the dashed red line. The error bars indicate the standard deviation predicted by the Gaussian process. In the upper left corner both the R^2-score and the mean absolute error (MAE) are shown with the error bars indicating the standard deviation computed based on three different CNNs.

An obvious approach to improve the performance is to increase the statistical representativity by aggregating the information of different images. We do this by averaging per material the predictions of all 40 images. In Figure 6.4 we plot the averaged predicted values against the target value for both the hardness and the carbon content. As is indicated by the relatively high R^2-score, the model correctly captures the tendencies in both cases. The lowest and highest values are typically over- and underestimated, respectively, which is to be expected as a Gaussian process is essentially an interpolation technique. However, the bulk of the data is accurately predicted, resulting in a relatively low mean absolute error in both cases. The standard deviations predicted by the Gaussian process indicate how uncertain the model is about

its predictions. As is indicated by the error bars in the figure, the standard deviations range from 10 HB to 50 HB for the hardness and from insignificant to 0.01 wt. % for the carbon content. In order to further improve the model, it would be best to add materials with a microstructure that is similar to the microstructures for which the uncertainty on the predictions is high.

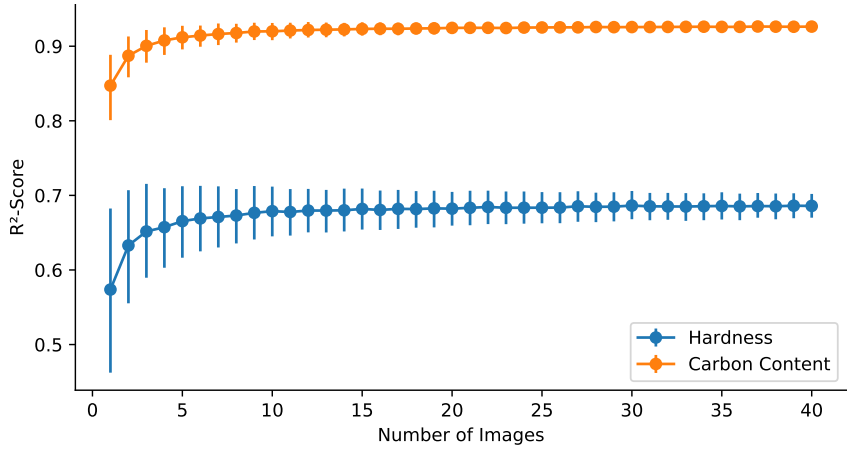


Figure 6.5: The R^2-score as a function of the number of images over which the predictions are averaged both for the hardness and the carbon content. The error bars are obtained by bootstrapping a 1 000 times for each number of images.

While the results shown in Figure 6.4 are promising, they are obtained by averaging the predictions of 40 images per material. The requirement of such a high number of images might hamper the practical adoption of our methodology. We therefore study the effect of the number of images over which the predictions are averaged in Figure 6.5. The figure clearly shows that averaging over more images always results in more accurate predictions, as we would expect. At the same time, it is clear that the gains of having more images are diminishing. We see that on average for about five images the performance is already close to what was illustrated in Figure 6.4. This suggests that the proposed methodology can safely be used when only about five SEM images per material are available. It is interesting to notice that the error bars for the hardness are much larger than those for the carbon content. Clearly, there are some images for which it is very difficult to accurately predict the hardness. This is most likely related to the results in Table 6.1 which suggest that the hardness cannot be predicted as accurately for images with a high magnification.

To assess the goodness of fit of our results, we compare the structure-

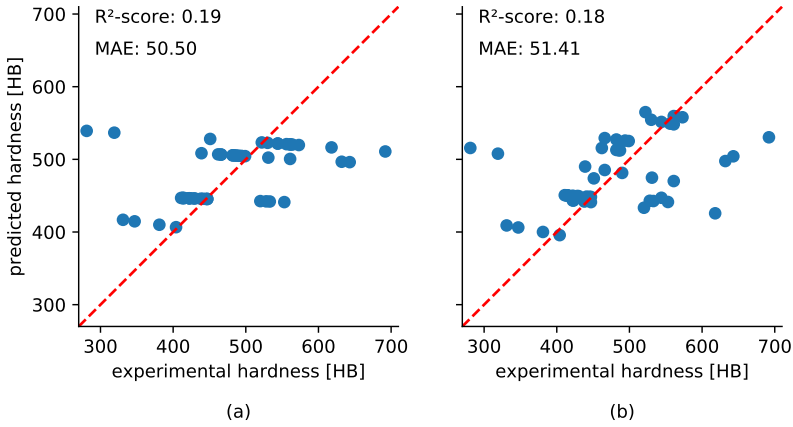


Figure 6.6: (a) The predicted hardness based on the carbon content using a Gaussian process. (b) The predicted hardness based on the carbon, silicon and manganese content using a Gaussian process. All predictions are obtained using leave-one-out cross-validation. In the upper left corner both the R^2-score and the mean absolute error (MAE) are shown.

property model to a model that predicts the hardness using information about the composition. In Figure 6.6 (a) we show the prediction of a Gaussian process that takes as input only the carbon content. Although the carbon content is usually considered to be a good predictor for the hardness [96], the obtained fit is clearly much worse than the structure-property predictions shown in Figure 6.4. When we include information about the manganese and silicon content, as is shown in Figure 6.6 (b), the majority of the points lie closer to the diagonal. However, because of outliers this is not reflected in the metrics. The presence of these outliers can be understood by considering that the silicon and manganese content only affect the hardness for specific thermal processing[97]. Because the model does not include any information about the thermal processing, some tendencies in the data cannot be explained. Overall, we find that the structure-property model performs better than the compositional models. The higher performance of the structure-property model indicates that the microstructural representations contain a lot of information about both the composition and the thermal processing. Further improving the structure-property relation will require larger datasets that contain many more materials, so that it becomes easier to generalize to new materials.

6.5 Conclusion

In this chapter we illustrated how representations from a CNN classifier network can be used to model the links between the composition, the structure and the hardness. We examined both the CNN representations discussed in chapter 4 and intermediate output from the model. We found that the models using the intermediate output yield more accurate predictions. We showed that based on the microstructure the carbon content can be accurately predicted even if only a single SEM image is used. Images at lower magnifications seem to work better, as they are more representative for the entire material. By combining the information of several images, the predictions improve and the hardness can also be accurately predicted. The presented structure-hardness models outperforms models that use information about the composition to predict the hardness.

7

Image resolution enhancement using deep learning

7.1 Introduction

From the previous chapters, it is clear that deep learning models work best when there is a lot of data available. Unfortunately, data can be scarce and time-consuming to obtain. In this chapter, we investigate whether it is possible to artificially increase the amount of training data by letting a deep learning model learn to increase the spatial resolution of SEM images, which is also called super-resolution. The main concern of this approach is the faithfulness of the enhanced images. Do they correspond well to reality or is the deep learning model too creative at enhancing the resolution of the image? Work done by de Haan *et al.*[98] suggests that a deep learning network can reliably increase the spatial resolution of SEM images of gold nanoparticles on carbon. However, we consider a more challenging problem, where we want to increase the resolution of SEM images of complex martensitic steels. This is more challenging, as we know from the previous chapters that the microstructures of this type of steel contain many small, yet important microstructural features. Two different models are trained: one to enhance the spatial resolution by a factor of two and one to increase it by a factor of four. We study the reliability of the enhancements both for materials on which the network is trained and for new, unseen materials. We also investigate whether the artificially obtained images can be used in conjugation with other machine learning models.

7.2 Methodology

Traditionally, the term super-resolution is used for methods that aim to reconstruct a high-resolution image from a set of low-resolution images that have subpixel misalignments.[99] Given a sufficient number of low-resolution images, it is possible to correctly retrieve the high-frequency content. This type of methods has already been successfully employed to obtain SEM images at very high magnifications.[100] It is also possible to perform super-resolution when only a single low-resolution image is given. This family of super-resolution methods is sometimes referred to as Single Image Super-Resolution (SISR).[101] Another important category of super-resolution methods are the so-called example-based super-resolution methods which learn a correspondence between low- and high-resolution image patches from a database.[102] Unlike the classical super-resolution methods, these example-based methods perform resolution enhancement to create visually pleasing images that do not necessarily contain the correct high-frequency content. Deep learning models have been extensively studied to perform example-based super-resolution.[103] The deep learning model takes a low-resolution image as input and returns a high-resolution version of the image. This enhanced version should be as close as possible to the given high-resolution image, which serves as the target of the deep learning model. Roughly two approaches are available to train super-resolution models. The first approach only requires a set of high-resolution images. These images are then subjected to degradation by downsampling and blurring the image to obtain a low-resolution version of the image. The deep learning model is then trained to restore the original high-resolution image from the low-resolution version. This type of methods hence only requires a set of high-resolution images and is therefore highly data-efficient. The disadvantage of this approach is that the performed image degradation does not necessarily lead to realistic low-resolution images, so that the deep learning model cannot be reliably used to increase the spatial resolution of low-resolution images.

A second approach requires pairs of low- and high-resolution images of the same region of the material. The deep learning network is given the low-resolution version of the image and should return a super-resolution version that resembles the high-resolution image as closely as possible. The disadvantage of this approach is that it requires much more data, because pairs of low- and high-resolution images are needed. The advantage, however, is that the deep learning models are assured to reliably enhance the resolution of the low-resolution images. We therefore will resort to this approach. By using transfer learning and data augmentation, we attempt to make the method as

data-efficient as possible.

One of the main difficulties of using pairs of low- and high-magnification images, is that they need to be perfectly aligned. To assure that this is the case, we use a two-step approach. In the first step, we use template matching[104] to obtain a rough alignment of the images. In template matching the high-resolution image is moved over the low-resolution image and for each position a similarity metric, for instance the negative mean squared error of the pixel values, is computed. The image is then aligned at the position with the highest similarity. This procedure only considers translations resulting in a coarse alignment of both images. The alignment is further optimized in the second step by using the ECC image alignment algorithm.[105] This algorithm identifies an affine transformation between the low- and high-resolution images by performing gradient descend using the correlation coefficient between the pixel values of both images as an objective function. This algorithm yields a better alignment, as it considers all possible affine transformations. A necessary condition for the algorithm to work well is that a good initial guess of the alignment is provided, which is why we first perform template matching.

In the work of de Haan *et al.*[98] a Generative Adversarial Network (GAN)[106] is used. This type of model consists of a generator network that outputs the super-resolution images and a discriminator network that tries to tell the difference between the super-resolution image and the high-resolution image. The generator and discriminator are competing with each other, where the generator tries to deceive the discriminator and the discriminator tries to detect the fake images created by the generator. Through this competition, both the generator and the discriminator become better at their task and we end up with a generator network that is able to produce very realistic high-resolution images. The advantage of this method is that there is no need to specify a loss function. However, the disadvantage is that the training procedure is not very stable as two networks need to be optimized at the same time.[107] This can be especially problematic when using larger network architectures. We will therefore resort to a different approach. Rather than using a discriminator network to express the difference between the real and the generated images, we use the perceptual loss as cost function.[108] This loss contains several terms. The first term is the mean absolute difference between the pixel values. The second term and third term are the feature and style reconstruction losses respectively and are based on the intermediate outputs of a pretrained VGG16 network. In a certain sense, one could argue that we have replaced the discriminator in the GAN by the pretrained VGG16 network. As we keep the parameters of this network fixed throughout the training, our approach is numerically more

stable than standard GAN training.

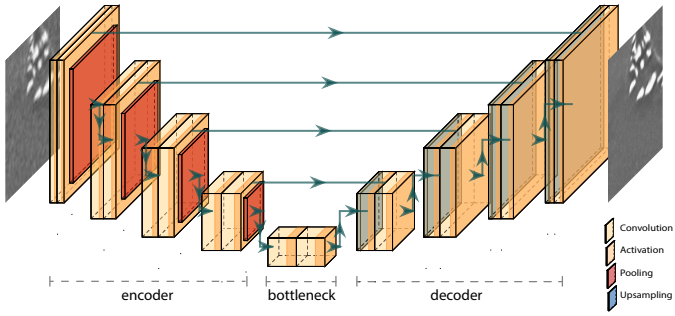


Figure 7.1: A schematic depiction of the Unet architecture used for super-resolution.

The Unet architecture we use to generate super-resolution images is shown in Figure 7.1. The encoder part of the Unet has a ResNet34 architecture.[70] For simplicity, however, we have depicted a VGG-like architecture in the figure. Throughout the encoder, the spatial dimensions of the image are gradually reduced, while the number of image channels is increased. The decoder part does the inverse operation and systematically increases the spatial dimensions until the initial image size is reached. The upsampling layers used in the network are pixel shuffle layers.[60] Characteristic to the Unet, there are shortcut connections that connect the intermediate outputs of the encoder to the intermediate inputs of the decoder layers. This greatly facilitates the optimization procedure. An advantage of the Unet architecture is that it is fully convolutional, i.e. there are no densely connected layers present. Because of this, the network is able to deal with input images of arbitrary size. To train the network, we make use of crops with a pixel resolution of 200×200. The parameters of the encoder are pretrained on ImageNet.[68] The training is done in two stages. In the first stage, only the parameters of the bottleneck and decoder layers are optimized, while the parameters of the encoder are kept fixed. In the second stage, all parameters are jointly fine-tuned at a lower learning rate. We use the Adam optimizer to optimize the parameters.[64]

The model architecture requires that the input and output images have the same size. As the low-resolution images normally have a smaller size, we first need to upsample these images. We find that during the training using bilinear interpolation leads to the most stable optimization. However, bicubic interpolation results in sharper images during the validation phase. The latter is unsurprising as bicubically upsampled images tend to be much sharper than their bilinearly upsampled counterparts. We conjecture that the

possible presence of interpolation artefacts in bicubically interpolated images perturbs the training phase of the network, making this interpolation scheme less suited than bilinear interpolation during training.

We use two methods to assess the quality of the super-resolution images. The first method is visually inspecting the artificially enhanced crops and comparing them to their high-resolution versions. We also compare the super-resolution images to the bicubically upsampled ones, which serve as input to the network. To evaluate the quality of the images more quantitatively, we assess how well the super-resolution images can substitute their high-resolution counterparts in a microstructure recognition task. Specifically, we train a random forest classifier on the Haralick features of a set of high-resolution images and evaluate the accuracy using ten-fold cross-validation. In each fold, we omit one of the ten high-resolution images for each material and train the random forest on the remaining nine images. The accuracy is evaluated on the omitted images. We also evaluate the accuracy for the low-resolution and super-resolution versions of these images. As the random forest is trained on high-resolution images, we would expect the accuracy to be lower for the low- and super-resolution images. However, if the network has learned to reliably enhance the resolution, the performance of the super-resolution crops should be close to that of the high-resolution ones. We also compare the accuracy of the super-resolution crops to that of conventional interpolation schemes such as bicubic upsampling and Lanczos upsampling.[109] We choose this approach to assess the image quality as it not only gives us one clear performance indicator on the image quality, the classification accuracy, but it also gives an indication of how well super-resolution crops can be used in conjugation with other machine learning models.

7.3 Datasets

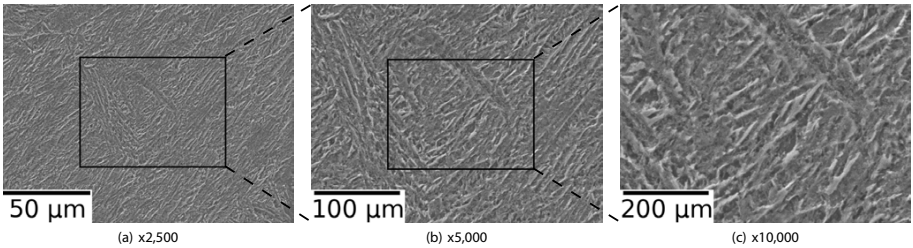


Figure 7.2: Some example images of the dataset. For the some area of the material, we have images at three different magnifications.

The dataset consists of 1 710 FEG SEM images with a pixel resolution of 1000×1200 of 57 different complex martensitic steels. For each material, we have ten images at three different magnifications: $\times$2 500, $\times$5 000 and $\times$ 10 000. Some example images are shown in Figure 7.2. For the evaluation of the quality of the super-resolution images of new materials, we randomly omit 10 materials from the dataset prior to the training of the Unet. We refer to this set of materials as the "unseen materials". The Unet is then trained on the remaining 47 materials. Once the Unet is trained, we randomly select 10 materials from the remaining 47 materials to train a random forest classifier on. This set is kept fixed and is referred to as the "seen materials", as these materials were already seen by the Unet during the training. We only train the random forest on 10 seen materials in order to be able to compare the classification accuracies of the seen and unseen materials in an unbiased manner, which would not be the case if the number of classes for both sets would be too different. During the training of the network, we randomly sample 50 crops with a pixel resolution of 200×200 per image in every epoch.

7.4 Results

In Figure 7.3, we compare a few crops that were upsampled by a magnification factor of two to their high-resolution counterparts. We see that the three versions of the same crop closely resemble each other. The edges in the super-resolution crops are somewhat sharper than those in the bicubically upsampled crops. At the same time, the super-resolution crops seem to be less pixelated as can be seen for instance in the upper left corner of the crop in the first row. The texture of the super-resolution crops appears to be much smoother than that of the bicubically interpolated one. We suspect that the pixelated regions in the bicubic upsampling are interpolation artefacts, as they were not present in the original low-magnification images. We can conclude that the Unet has learned to distinguish between important microstructural features and artefacts and that it is able to enhance low-resolution images somewhat better than bicubic upsampling does.

For the crops whose magnification is increased by a factor of four, there is more difference between the upsampling methods, as is shown in Figure **??**. The super-resolution images are clearly much sharper and more detailed than the crops obtained through bicubic upsampling, which are highly pixelated and blurred because of the fourfold upsampling. Once more, we see that the Unet manages to recognise the important microstructural features, as it greatly accentuates the edges in the image. However, if we compare the tex-

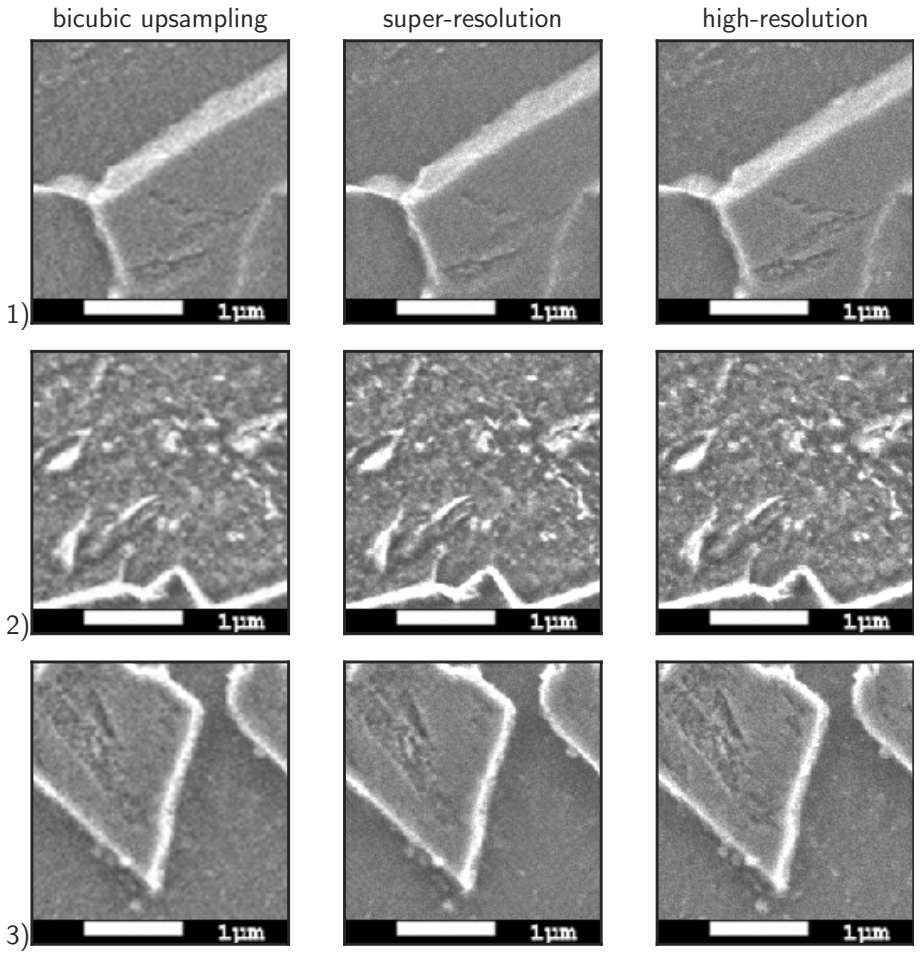


Figure 7.3: A comparison of the bicubically upsampled low-resolution crops, which serve as the input to the deep learning, the super-resolution crops generated by the Unet and the high-resolution crops. All crops belong to the set of unseen materials and are upsampled by a factor of two.

ture of the super-resolution crop in the last row to that of its high-resolution equivalent, we see that there are some small differences as is indicated by the arrow. This suggests that the Unet does not always reliably enhances the resolution of the image. Given that the highly blurred crops in the first column are the input to the Unet, it is understandable that Unet does not always have sufficient information to reliably resolve small microstructural features in the image.

To analyse the results more quantitatively, we train two random forest

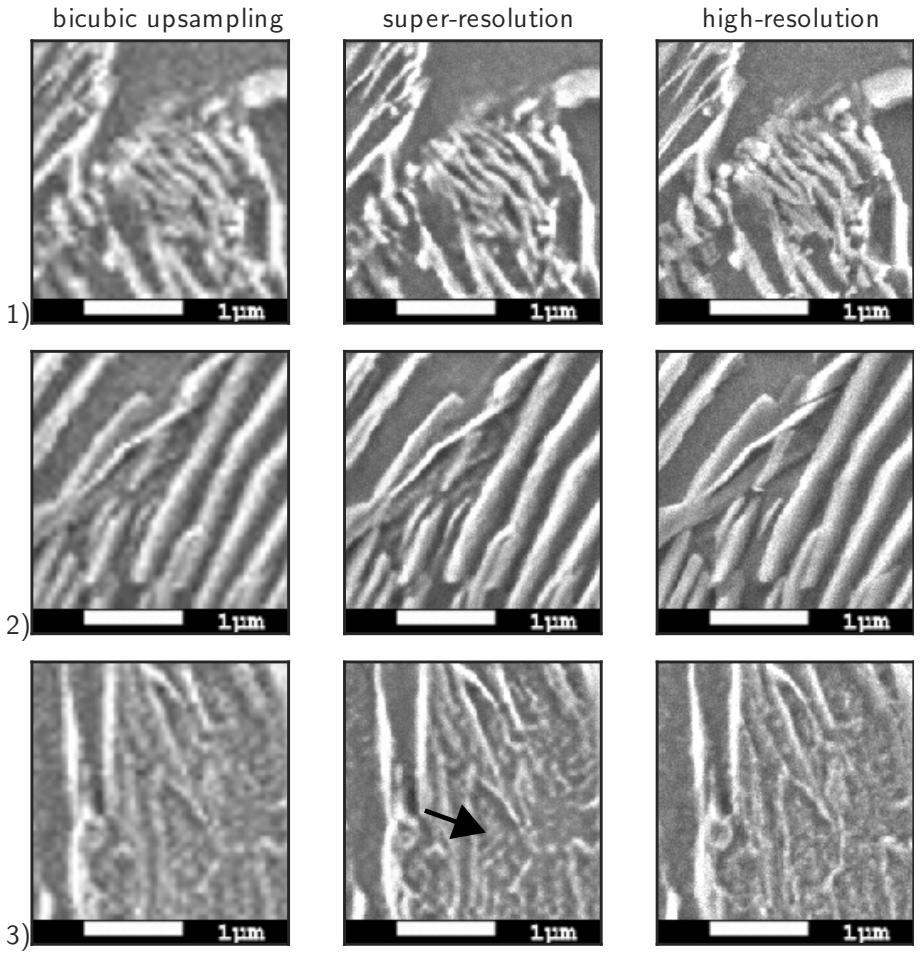
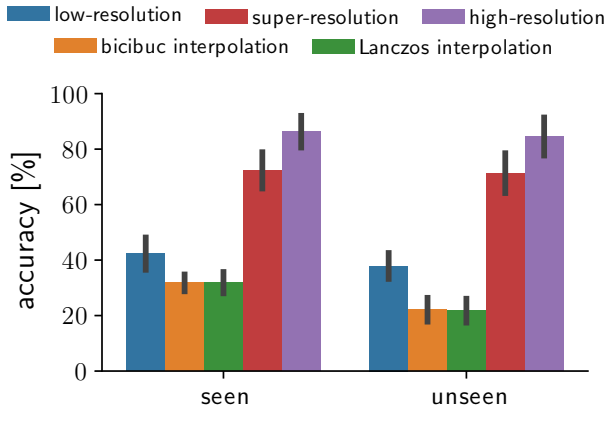
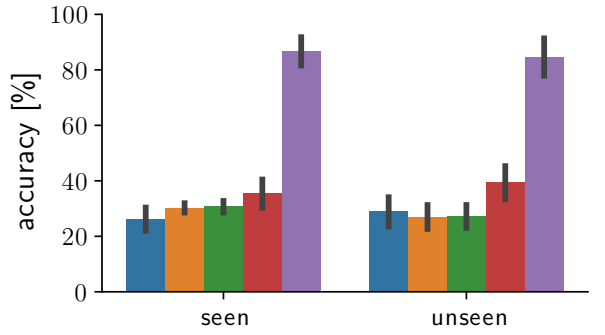


Figure 7.4: A comparison of the bicubically upsampled low-resolution crops, which serve as the input to the deep learning, the super-resolution crops generated by the Unet and the high-resolution crops. All crops belong to the set of unseen materials and are upsampled by a factor of four. The arrow indicates a region where the deep learning model generates a texture that is different from the high-resolution version.

classifiers on the high-resolution crops. One classifier for the ten randomly selected seen materials and one for the ten unseen materials. In Figure 7.5, we compare for both magnification factors the accuracies for the different types of upsampled crops. For the high-resolution images, we obtain accuracies of around 85 % for both the seen and unseen materials. This relatively high accuracy indicates that Haralick features are very suited to describe



(a) magnification factor 2



(b) magnification factor 4

Figure 7.5: A comparison of the classification accuracies of low-resolution, bicubically upsampled, Lanczos upsampled, super-resolution and high-resolution crops for both magnifications factors and for both the seen and unseen materials.

the texture of complex martensitic microstructures. The accuracy of the low-resolution crops for twofold magnification model is much lower and only around 40 %. As the low-magnification crops contain considerably less detail then their high-resolution counterparts this was to be expected. The conventional interpolation schemes obtain an accuracy that is even lower. These schemes follow a standard recipe to perform the upsampling and thus do not add any useful information to the image. On the other hand, the super-resolution crops achieve an accuracy of around 75 %. This suggests that for the twofold magnification, the neural network adds useful information to the low-magnification crops and performs a reliable enhancement of the resolution. Furthermore, this result indicates that spatial resolution enhancement can be used in conjugation with other machine learning methods.

For the fourfold magnification model, the same conclusions hold for the low-magnification crops and the crops that were upsampled with either bicubic or Lanczos upsampling. However, for the super-resolution crops, we see that the accuracy is now barely any better than that of the other upsampling methods. This is surprising, as during our visual inspection we found that the sharpness and amount of detail in these crops is significantly better than in their conventionally upsampled counterparts. This suggests that the Unet does not perform a reliable resolution enhancement in this case.

A last interesting thing to remark in Figure 7.5 is that the difference in accuracy between the seen and unseen materials is negligible for the super-resolution crops. This suggests that the trained Unet generalizes well to new complex martensitic steels. The twofold magnification model can therefore be a practical tool in the study of this type of materials. As a twofold increase in magnification leads to a fourfold decrease in characterization time, such a tool can lead to great time and cost savings. If we could train a similar model on a larger, more diverse dataset of microstructure images, it would be possible to obtain a model that can enhance many more types of microstructures.

7.5 Conclusion

In this chapter we presented a method to artificially enhance the spatial resolution of SEM images of complex martensitic steels using deep learning. We studied two models: one to perform a twofold increase in magnification and one to perform a fourfold increase in magnification. For both magnification factors, we visually analysed the image quality of the enhanced crops. Both models produced high quality crops and they have clearly learned to recognise important microstructural features in the image. Especially the model that performs a fourfold increase in magnification generated crops that are significantly sharper and more detailed than interpolated images. However, when we analysed how well the super-resolution crops can substitute the high-resolution ones for a microstructure recognition problem, we found that the fourfold interpolation model is barely more useful than interpolation schemes. We therefore concluded that the presented method is unsuited to perform fourfold magnification reliably. For the twofold increase in magnification, on the other hand, we find that the deep learning model provides us with reliably enhanced crops that can be successfully used in combination with other machine learning methods. The model also generalized well to new materials, so that the model can be used as a practical tool to enhance the resolution of complex martensitic microstructures.

8

Conclusions and perspectives

πάντα ῥεῖ
Heraclitus

8.1 Conclusions

In this PhD we studied the possibilities of machine learning for metallurgy. The central question was how we can extract more information from microscopy images and what we can do with this information. Given the incredible performance of deep learning models for many computer vision tasks in recent years, we mainly focussed on deep learning methods. These methods have the advantage that they directly take an image as input, rather than features that are hand-crafted by domain experts. The deep learning model learns for itself which features in the image it deems relevant for the task at hand. Because of this, the features are learned directly from the data without relying on any kind of domain knowledge. While domain knowledge is often useful and even crucial, a purely data-driven approach could offer complementary insights. The disadvantage of such a data-driven approach is that it requires large datasets. As in material science large datasets are scarce, a lot of emphasis was put on making existing deep learning methods applicable to smaller datasets containing only a few hundreds of microscopy images. Table 8.1 gives an overview of the different datasets that are used in this work.

chapter	usage	type image	#images	#classes
4	training+testing	OM	778	60
	zero-shot	OM	30	10
5	training quiz 1	OM+SEM	134	5
	quiz 1	OM+SEM	36	6
	training quiz 2	SEM	58	6
	quiz 2	SEM	21	7
6	training CNN	SEM	884	26
	evaluation PSP links	SEM	2080	52
7	super-resolution	SEM	2280	57

Table 8.1: An overview of the different datasets that are used in this thesis.

The first problem we studied was how deep learning can be used to represent microstructure images. Concretely, we wanted to extract a limited set of numbers from a microscopy image while retaining as much information about the microstructure of the material as possible. We proposed triplet networks as a means to achieve this goal. Triplet networks are a deep learning method that optimize the image representations so that the distance in the representation space is a similarity measure. If two microscopy images have representations that lie close to each other, they will therefore be visually very similar. We demonstrated that by using triplet networks it is possible to obtain low-dimensional representations that contain a lot of information about the microstructure. We found that when only extracting two or three numbers from a microscopy image, it was already possible to differentiate between materials that had only minor differences in processing and composition. Furthermore, the representations are naturally clustered together in groups of similar materials, which implies that the learned similarity measure corresponds well to the human notion of similarity. We also compared the triplet representation to other microstructural representations that are commonly used in the literature. We found that triplet representations can better differentiate between different materials than other methods, despite the fact that they are much lower in dimensionality. When we would like to use these microstructural representations as input for a machine learning model, the low dimensionality of the representations offers a great advantage as it leads to much more robust and data-efficient machine learning models. A disadvantage of triplet network representations is that they do not generalize well to new materials on which the deep learning network was not trained. The representations of a convolutional neural network that is trained on a microstructure recognition task using the cross-entropy loss generalized somewhat better to new materials, which is also reported in the literature. In

any case, we could safely conclude that deep learning holds great potential in representing the microstructure.

Triggered by the good results of the triplet network representations in differentiating between very similarly looking materials, we decided to organise a competition in which the triplet network would compete against expert metallurgists in recognising microstructures. The competition consisted of two quizzes in which the participants had to assign a number of images to one of the predefined material classes. The first quiz contained both OM and SEM images of different types of microstructures. The second quiz only featured SEM images of complex martensitic steels. An additional difficulty was that in the first quiz it was possible that the material on the quiz image did not belong to any of the predefined classes. To deal with this, we used Gaussian mixture models to fit a distribution to the microstructure representations obtained by the triplet network. Because of this distribution, we could assign a likelihood to each microstructure. If the likelihood was too low, the microstructure was considered to be an outlier that does not belong to any of the predefined classes. Using this methodology, we could obtain a higher score than the average expert for both quizzes. For the second quiz, the deep learning model even obtained a perfect score. Although it was already known from other fields such as medicine that deep learning could beat domain experts in image recognition tasks, this result was surprising as only a handful of images per material class was used to train the model. This suggested that we have managed to obtain a methodology to train high performance deep learning models in a very data-efficient manner. We therefore concluded that large datasets are not necessarily required to obtain powerful deep learning models, although having more data will always result in better models.

One of the central aims of this PhD was to investigate whether machine learning can help to understand the relation between the composition, the structure and the properties of steel. More specifically, we focussed on understanding the relation between the composition, the microstructural information in SEM images and the hardness for a set of complex martensitic steels. In order to obtain good microstructural representations, we started by training a convolutional neural network on a microstructure recognition task on a dataset of similar steels. We trained a convolutional neural network using the cross-entropy loss rather than a triplet network, as we found that the representations of the former generalized better to new materials. Upon training a Gaussian process that takes these representations as input, we found that the CNN representations only have a limited explanatory power for both the composition and the hardness of the material. We conjectured that this is because the CNN representations are optimized for

microstructure recognition and are too task-specific. To obtain more general microstructural representations, we extracted intermediate output from the convolutional network. As this output was too high-dimensional, we applied PCA and retained only the ten most important components. Using these, we could accurately predict the carbon content based on a single SEM image. By combining the information from different images, we could also obtain accurate predictions for the hardness, as is shown in Figure 8.1, and for the manganese content. Although the obtained fits are not perfect, our results clearly indicated that it is possible to extract valuable information about the composition and the properties of the material based on microscopy images.

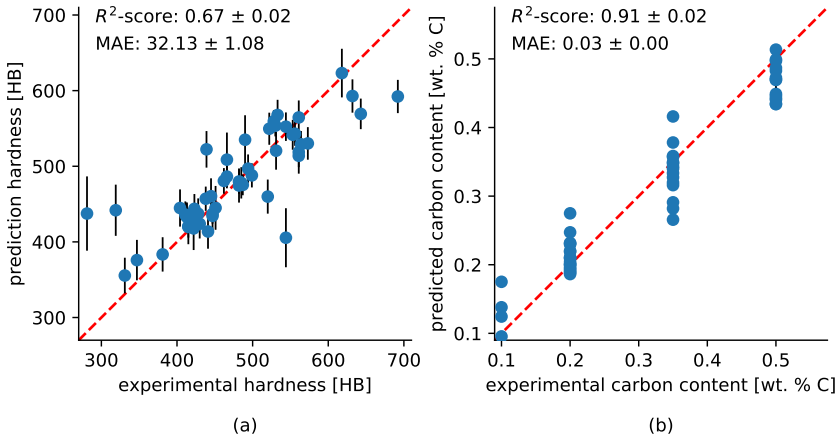


Figure 8.1: A scatter plot of the target values against the predicted values for both the hardness (a) and the carbon content (b) using the average prediction of all 40 images per material.

A last application of deep learning we studied, was the artificial enhancement of the spatial resolution of SEM images of martensitic steels. As deep learning models perform better when larger datasets are used, such a resolution enhancement could be a cost effective method to increase the size of datasets. We studied the Unet architecture as a method to achieve either twofold or fourfold magnification for a given low-resolution image. Upon visual inspection of the enhanced images, we found that especially the fourfold magnification model generated images that were significantly sharper and more detailed than those interpolated by conventional interpolation schemes. However, because the low-resolution input images were extremely blurry, there were some small distortions in the enhanced images. When used in conjugation with another machine learning model that is

trained on a microstructure recognition task, these small distortions proved to be detrimental for the performance of the classification model. We thus concluded that deep learning is not suited to reliably perform fourfold magnification. For the twofold magnification, we did find that the Unet performs resolution enhancement much more reliably than conventional interpolation schemes and that it can be safely used in combination with other machine learning methods. Thus, we have demonstrated how deep learning can not only be used to analyse large amounts of data, but how it can also be used to generate even more data.

Based on the results presented in this work, there is only one real conclusion to draw: the potential of deep learning in the analysis of microstructure images is vast and we have only scratched the surface.

8.2 Perspectives

At the start of this PhD we had access to a GPU with a RAM memory of 12 Gb. Now, about three and a half years later, most calculations are performed on GPUs with a RAM memory of 32 Gb. Furthermore, these GPUs are much faster and require less memory for the same operations than the GPUs we initially used. The performance of the hardware that is used to train deep learning models is increasing at a staggering speed, enabling the usage of deeper architectures, larger batch sizes and more sophisticated optimization algorithms. Furthermore, if one considers how much the deep learning methods and architectures have improved since it gained major public interest in 2012, we anticipate many more exciting breakthroughs on the methodological side in the years to come. Lastly, the relentless pace at which datasets are increasing in size, yields great prospects for building improved deep learning models. We identify these three reasons, i.e. better hardware, better methods and bigger datasets, to safely state that the models that were obtained in this PhD will be replaced by much better ones within the next two to five years. Given the very promising results we were able to present in this work, we can only be excited about the role deep learning will play in metallurgy in the years to come. Although it is very difficult to predict what the future will bring, we will try to point out some promising directions based on the results obtained in this work.

One of the key shortcomings in deep learning is the lack of model interpretability. In this work, we proposed a simple and effective way to construct saliency maps to get an idea of which microstructural features the deep learning model deems important. Throughout this PhD, we have experimented with many different methods to create reliable saliency maps,

but unfortunately most methods did not yield clear results. Even with the presented method, it still took considerable effort to understand what the network looked at and it is clear that there is still a lot of room for improvement. Model interpretability is important for two reasons. The first one is that it allows us to trust the model predictions. The performance of deep learning model can be sometimes so incredible that you need to be sure that the model is not looking at unwanted features such as artefacts due to either etching or image compression, differences in the illumination conditions or others differences in the characterization procedure. Only by constructing reliable saliency maps, we can be sure that the model has understood what the relevant features in the images are. The second reason why model interpretability is important is for the practical application of deep learning. While it is clear that deep learning can outperform experts, it is unlikely that it will immediately replace those experts. Rather, we expect that deep learning model will initially become a valuable second opinion for the experts in the decision-making process, as is already the case in some fields of medicine. In order to take the right decisions, it is necessary that the experts understand why the deep learning model makes certain predictions. Better model interpretability will therefore be the key towards practical adoption of the methods we presented in this work. Fortunately, a lot of research is done in this direction and we firmly believe that in the next couple of years deep learning models will become less of a black box.

Closely related to the topic of interpretability, is the issue of uncertainty quantification. Deep learning methods and machine learning models in general are very bad at extrapolating. This is something we also clearly saw when we tried to use the triplet representations to recognise new materials. It is unavoidable for data-driven models that they become unreliable for new data that is significantly different from the data the model was trained on. However, it is possible for the model to quantify how similar new data is to the training data and to provide the user with a level of uncertainty on the prediction. This is one of the reasons why we preferred to use Gaussian processes to establish the PSP links. These models naturally provide us with a reliable uncertainty estimate. Most deep learning models however, only make a prediction without revealing any information on how certain they are. By replacing the parameters of a deep learning model by probability distributions, it is possible to do statistically sound quantification of uncertainty. However, doing so results in networks that are much harder to optimize and most of these probabilistic methods do not scale to practical problems. The advent of better hardware and better optimization methods will without a doubt lead to scalable approaches to train such probabilistic deep learning models.

We only studied supervised methods in this work: the machine learning models had a clear target value to which the prediction should be as close as possible. Because of the clear target value, supervised learning is relatively easy from a methodological point of view. However, it requires labelled data, which can be extremely scarce as was for instance the case for the structure-property prediction. Unsupervised methods are much more data-efficient, but can be notoriously difficult to optimize. In recent years, a lot of research has been devoted to unsupervised methods such as auto-encoders, where a deep learning model has to compress the data in the image, and deep clustering, where a deep learning model clusters similarly looking images based on some metric. Without a doubt, these methods can prove to be very useful to learn microstructural representations. Unsupervised methods have the additional advantage that they tend to generalize better to new data or new materials. During our research, we found that the performance of unsupervised methods is still inferior to supervised learning, but as the methods improve, we are convinced that they will play a crucial role in making deep learning methods more data-efficient.

A last important evolution is the public availability of data. One of the key drivers of the rapid progress in deep learning has been the public availability of the ImageNet dataset. This dataset serves as an important benchmark for new methods and almost all pretrained networks are trained on ImageNet, as is also the case for the networks used in this work. Therefore, the importance of public available datasets like ImageNet cannot be overstated. We are convinced that it would greatly benefit research in deep learning for the analysis of microstructures if large datasets of microstructure images were made available. We applaud efforts made by DeCost *et al.*[110] to share their dataset. We hope that many other independent research group will follow their example.

We sincerely hope that through this work we have managed to arouse the reader's interest in machine learning in metallurgy. We firmly believe that this work constitutes a solid foundation for further research into the different topics that were covered in this text.

Part II

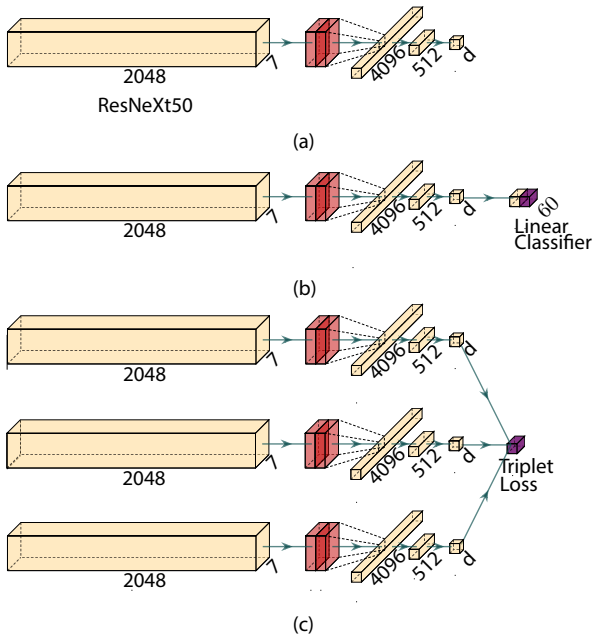
Published Papers

A

Publications in International Peer–Reviewed Journals

Paper I

Compact representations of microstructure images using triplet networks



M. Larmuseau, M. Sluydts, K. Theuwissen,
L. Duprez, T. Dhaene and S. Cottenier

NPJ Computational Materials, **2020**, *6*, 1–11

M. Larmuseau conceived the methodology, wrote the software, performed the calculations, interpreted the results and prepared the manuscript.

npj Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

Compact representations of microstructure images using triplet networks

Michiel Larmuseau[1,2,3✉], Michael Sluydts[1,2], Koenraad Theuwissen[4], Lode Duprez[4], Tom Dhaene[3] and Stefaan Cottenier[1,2]

The microstructure of a material, typically characterized through a set of microscopy images of two-dimensional cross-sections, is a valuable source of information about the material and its properties. Every pixel of the image is a degree of freedom causing the dimensionality of the information space to be extremely high. This makes it difficult to recognize and extract all relevant information from the images. Human experts circumvent this by manually creating a lower-dimensional representation of the microstructure. However, the question of how a microstructure image can be best represented remains open. From the field of deep learning, we present triplet networks as a method to build highly compact representations of the microstructure, condensing the relevant information into a much smaller number of dimensions. We demonstrate that these representations can be created even with a limited amount of example images, and that they are able to distinguish between visually very similar microstructures. We discuss the interpretability and generalization of the representations. Having compact microstructure representations, it becomes easier to establish processing–structure–property links that are key to rational materials design.

npj Computational Materials (2020)6:156; https://doi.org/10.1038/s41524-020-00423-2

INTRODUCTION

Every material has its own combination of physical properties such as hardness, toughness and ductility. These properties determine the industrial applications for which the material is suited. By improving the properties of the material, its performance for the application increases, potentially leading to higher efficiencies, longer lifetimes and an overall reduction in costs. Thus, the discovery of new materials with enhanced properties is a key driver of technological advancement. To discover new materials in a systematic way, it is necessary to model the properties of the material, corresponding to a given composition and set of processing parameters. Although it is possible to directly link these parameters to the properties of the material, there are two main drawbacks to this approach. First, data on properties of tailor-made metals is scarce, as these metals are expensive to produce in small quantities. This makes it difficult to build robust models. Second, the obtained model would be highly specific for the industrial equipment that is used to make the metals. Furthermore, it is known that the structure of the metal at a small scale, the microstructure, determines the properties of the material and in its turn is strongly affected by the composition and processing conditions[1]. Therefore, the microstructure, which is typically characterized by a set of microscopy images of prepared surfaces of the metal, is an essential link between the processing and the properties of the material. In the literature, this is designated as processing–structure–properties links[2,3]. Although images are an important source of microstructure information, raw image data are too complex to directly link to properties. For this reason, the information in the images must be condensed in a more simple, compact representation.

The question of how a microstructure can be best represented has been investigated by metallurgists for decades[4]. In the past years, inspiration has come from the field of computer vision, where the introduction of machine learning methods has led to the development of performant representations[5–8]. However, all these representations are obtained by applying fixed procedures to the available images. No microstructure-specific information is used. Herein, we investigate whether it is possible to further improve the representations by using a machine learning algorithm to learn how a microstructure can be best represented from the available data. As increasingly large microstructure databases are becoming available, such a data-driven approach seems promising. From the field of deep learning, we propose triplet networks as a method to learn optimal representations directly from the available microstructure data[9,10]. These representations have two desirable properties. First, a distance between two data points can be defined and can be used as a similarity measure: visually similar microstructure images will have representations that lay close to each other in the representation space. Second, the dimensionality of the representation can be freely chosen. In order to build robust machine learning models, it is recommended that the number of model inputs remains small[11]. This implies that the microstructural representation should be preferably as low-dimensional, or compact, as possible. We investigate how many dimensions a microstructural representation needs to have in order to be able to encode sufficient detail about the material. The combination of these two desirable properties makes it possible to faithfully visualize many microstructures in a single plot.

Research in automated microstructure recognition has mainly focused on distinguishing between groups of materials with significantly different compositions and processing conditions and consequently clear visual differences. When examining the literature one quickly finds several recent examples, all of which report close to perfect performance[5,7,12,13]. However, the question remains whether it is possible for a machine learning model to learn to distinguish between materials with only minor differences in composition and processing. This question is highly important

[1]Center for Molecular Modeling, Ghent University, Technologiepark 46, B-9052 Zwijnaarde, Belgium. [2]Department of Electrical Energy, Metals, Mechanical Constructions and Systems, Ghent University, Technologiepark 46, B-9052 Zwijnaarde, Belgium. [3]IDLab, Department of Information Technology, Ghent University - IMEC, Technologiepark 126, B-9052 Zwijnaarde, Belgium. [4]OCAS NV/ArcelorMittal Global R&D Gent, Pres. J. F. Kennedylaan 3, B-9060 Zelzate, Belgium. ✉email: michiel.larmuseau@ugent.be

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

npj nature partner journals

for two reasons. First, being able to distinguish between materials with a different composition and processing is necessary to successfully establish a link between processing and structure. Second, as the properties of steel alloys are very sensitive to small changes in composition and processing conditions[14], detecting these changes is essential to establish a successful link between microstructure and properties. As the resulting microstructures are often visually similar, expert metallurgists also have difficulties recognizing these differences. Machine learning models could consequently prove invaluable in making these analyses faster and more accurate. In this study, we use a dataset of 60 materials, each characterized by their chemical composition and processing procedure. None of them is exactly identical to another material in the set; some of them have rather similar compositions and processing procedures, whereas others have very different ones. According to the regular metallurgical classifications, these 60 materials can be divided in 5 classes: pearlite with austenitic matrix, martensite with prior austenite grains, tempered martensite, quenched martensite and ferritic steel. In contrast to what is usually done in the literature, we do consider these 60 materials as independent ones. This allows us to examine to which extent the triplet network representations are able to discern differences between very similar materials. If that turns out to be the case—which it will—then the triplet network sees more information in the images than conventional metallurgy does.

RESULTS AND DISCUSSION

Microstructure representations used in the literature

We briefly discuss the most commonly encountered microstructural representations in the literature. These representations are used as a comparative benchmark for our method.

A first set of methods are inspired by the work of Chowdhury et al.[5]. The methods we evaluate include Haralick features, contrast features and local binary patterns. Haralick and contrast features extract information based on the distribution of the greyscale values of neighbouring pixels. Local binary patterns also consider the immediate neighbourhood of the pixels in the greyscale image and keep track of how many times each type of neighbourhood occurs. All these features therefore aggregate local information to obtain a global description of the image, typically stored as a numerical vector.

The visual bag of words approach used by DeCost et al.[15] inspires the second set of methods. A dictionary of commonly occurring visual keypoints is constructed and for each image the number of occurrence of each of these keypoints is counted. The size of the resulting feature vector depends on the number of common keypoints that are in the dictionary.

A third type of features is obtained by applying principal component analysis on the two-point statistic[8], which is essentially the auto- and cross-correlation function of the binary image. Only the most important principal components are retained, resulting in a compact description of the microstructure. An additional benefit is that this description allows for image reconstruction, as discussed in Fullwood et al.[8]. However, this is partially because the input image is reduced to binary values; hence, a lot of information is already omitted beforehand. As the two-point statistic is used, we expect the method to be mainly useful for materials that consist out of two distinct components such as dual phase steels.

We also examine the recently introduced translation-invariant texture features[7]. Here, a dictionary of visual words is created by aggregating the intermediate output of a Convolutional Neural Network (CNN). As described in the study, we apply three different encodings of this output (Vector of Locally Aggregated Descriptors (VLAD), mean and max encoding). We consider both the output of the last block and second to last block of a pretrained deep learning network with a VGG16 architecture[16].

Lastly, we have also included the features used in Gola et al.[6]. Here, a modified version of the Haralick features, called textural features by the authors, are used in combination with morphological features of the grains. We consider the performance of these textural and morphological features separately, but also use a genetic algorithm to select the most relevant subset of these features, as is described in Gola et al.[12].

We do not include results from image segmentation methods[13,17,18]. For these methods, the aim is not to assign a given microstructure image to the correct material class, but rather to assign each individual pixel of the image to the correct material class. Although such an approach clearly gives a more complete analysis of the image, it requires training data where each of the pixels is manually labelled by human experts. In case one considers a four class classification problem, such a labelling procedure is still doable, albeit labour intensive[13,17]. In our case, where we consider 60 different classes, manually assigning each pixel to one of these classes is nearly impossible. When microstructures become more complex, it is also harder for human experts to label the data consistently. We therefore do not include any methods that use pixel-labelled data in our benchmark.

Making microstructure representations with triplet networks

We present a deep learning model[9] that allows to construct low-dimensional representations of microstructure images. As a deep learning model can have several millions of trainable parameters, a large amount of data is required to train a network in such a way that the model is sufficiently general. As we only have a dataset with less than 1000 images, training such models from scratch is not an option. We therefore use a multi-stage approach, which gradually refines the representations starting from a pretrained model, as illustrated in Fig. 1. We modify the architecture of the pretrained convolutional network and finetune its parameters to the microstructure classification task. The resulting model then serves as the starting point of a so-called triplet network[10], which

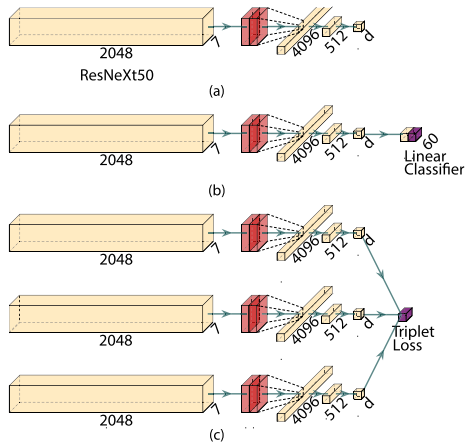


Fig. 1 Steps in the training procedure of the triplet network. **a** The output network of a pretrained ResNeXt50 model is replaced by bottleneck layers. **b** The output layers are fine-tuned to the microstructure classification task. **c** The representations are further improved by optimizing the triplet network.

will generate the final representations. Each step is discussed in detail in the following paragraphs.

As a starting point, we use the so-called ResNeXt50 architecture[19]. The output of the ResNeXt50 model is a 4096-dimensional vector, which clearly is too large for our purposes. To this end, we replace the final fully connected layers of the network, reducing the output dimensions to form a bottleneck, similar to the encoder network in an autoencoder[20]. An advantage of this architecture is that we can freely choose the dimensionality of the representation space, which we will denote as d.

To speed up the training of the triplet network, we start by fine-tuning the model on a classification task, where the model must learn to distinguish between the 60 different classes of materials present in the dataset. Fine-tuning a pretrained model is referred to as transfer learning[21]. In the literature, it is found that by using transfer learning, one can obtain models that are much more general and easier to train[22]. To perform the classification, an additional fully connected layer is added to the network. This layer linearly maps the representations onto a C-dimensional space, where C denotes the number of classes. After applying the softmax activation function, the probabilities of belonging to each of the classes is obtained. The negative log-likelihood cost function is used to optimize the network taking these probabilities as input. The probabilities can be interpreted as levels of confidence, providing a clear advantage over hard classification. The layers we have included for performing the classification task, can be seen as a logistic regression model. This model takes the d-dimensional representations of the images and applies logistic regression to classify these images. As logistic regression is a linear classification method, this implies that the model will only try to make the representations belonging to the different classes linearly separable. This is not necessarily a desired feature of a microstructural representation, as images of the same material can still be relatively remote from each other in the representation space.

A desired feature of a representation should be that 'similar' images, meaning images belonging to the same material, should be close to each other in the representation space. To this end, we propose to use triplet networks[10]. Triplet networks consist of a single deep learning network that maps each image $\mathbf{x}_i$ onto a representation vector $\mathbf{f}(\mathbf{x}_i)$. Rather than training with one image at the time, three images are used, of which one is used as reference, the anchor $\mathbf{x}_i^{\mathbf{a}}$, one is used as an example of the same class, the positive example $\mathbf{x}_i^{\mathbf{p}}$ and the final one is used as an example of a different class, the negative example $\mathbf{x}_i^{\mathbf{n}}$. The criterion used during the parameter optimization is the triplet loss[23]:

$$\sum_{i}^{N} \max\left(0, ||\mathbf{f}(\mathbf{x}_i^{\mathbf{a}}) - \mathbf{f}(\mathbf{x}_i^{\mathbf{p}})|| - ||\mathbf{f}(\mathbf{x}_i^{\mathbf{a}}) - \mathbf{f}(\mathbf{x}_i^{\mathbf{n}})|| + \alpha\right), \quad (1)$$

where N is the number of triplets used in the batch and α is a positive, real number that represents the margin between the positive and negative pairs. Intuitively, this loss will minimize the distance between the representations of microstructures belonging to the same class and maximize the distance between microstructures belonging to different classes. By using triplet networks, we thus introduce distance as similarity measure between images. The triplet network only needs to know which images belong to the same class and which images do not. This is an important difference with the CNN, where each material needs to have its own label as class information is explicitly encoded in the softmax layer. Because of this, we expect to more easily extend triplet models to new microstructures in the future, when applying this model on different datasets.

The datasets

Two datasets are used in this work. The first dataset contains 778 optical microscopy images of 5 visually different groups of austenitic, martensitic and ferritic steels, as is shown in Fig. 2a. Within these groups, we consider materials that have slightly different compositions and processing conditions, leading to a total of 60 different material classes. The ranges of compositions per group are listed in the Supplementary Table 1. For each material, images were taken at half and quarter thickness, at three different locations in the cross-section. All images are converted to greyscale.

In Supplementary Fig. 1, we show a typical microstructure image for each of the classes in the dataset. Some of these images appear to be almost identical, even to the trained human eye. For each class, there are 7 to 24 different pictures with a 1000×1200 pixel resolution. We express the magnification of the images in terms of the inter-pixel distance, which is the physical distance between two neighbouring pixels. For this dataset, the inter-pixel distances range from 0.1 to 5 µm. The number of images and the different magnifications for each class is shown in Supplementary Fig. 2.

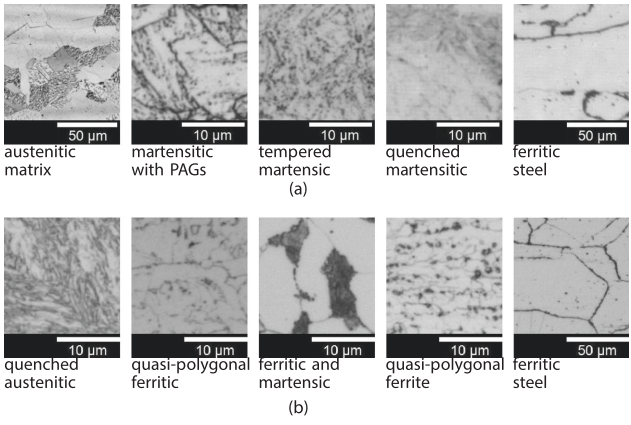


Fig. 2 Examples of the optical microscopy crops seen by the model. a For the first dataset, we show one crop for each group, which is described in more detail in Table 1. **b** For the second dataset, we show crops of the first five material classes.

The second dataset is smaller and is used to verify how well our model generalizes to unseen types of microstructure images. The variety of microstructure types is larger in this dataset and most of the microstructures commonly encountered in current industrial steels such as martensite, bainite and pearlite are included. It consists of thirty 1000×1200 pixel images belonging to ten different material classes. For each class there are three images, all with the same magnification. The inter-pixel distances range from 0.1 to 0.5 µm. Figure 2b shows a few examples of images belonging to different classes for the second dataset. An example for each of the material classes is shown in Supplementary Fig. 3 and short description of these classes can be found in the Supplementary Table 2.

Using the datasets described above, we first try to answer the question of how many dimensions we need to represent microstructure images, while still being able to distinguish between different materials. Second, we analyse the strengths and the weaknesses of the triplet model in more detail and investigate the visual features the model deems important using saliency maps. Next, we compare the discriminative power of the presented representations to other methods found in the literature. Finally, we check how general the representations are by applying the model to a microstructure recognition task with different materials. The main performance metric we use throughout this work is the accuracy, defined as

$$\text{Accuracy} = \frac{\#\text{Crops correctly classified}}{\#\text{Total crops classified}} \cdot 100\%. \quad (2)$$

We define the train accuracy as the accuracy on the dataset used to train the model and the test accuracy as the accuracy on the unseen test set, which was constructed using the procedure described in the "Methods" section. All accuracies reported in this section are obtained by training a random forest model on the representations under consideration[24].

Evaluating the accuracy for different representation sizes

One of the main advantages of the presented method is that it allows the user to choose the dimensionality of the representation. Although the ideal dimensionality might differ for other datasets, it still provides us with a valuable indication on how compactly we can represent a microstructure. In Fig. 3, we show the classification accuracy as a function of the representation dimension, for features obtained both from the CNN bottleneck layer and the triplet network for the task of recognizing the correct material class. In both cases, a random forest model[24] is used to perform the final classification. For a three-dimensional representation, we already see we can do better than all currently existing methods. The difference between the triplet network representations and the CNN representations is only significant for two and three dimensions, where the triplet network performs better. The result for a representation dimension of 512 indicates the performance in case no bottleneck layer is used. As expected, the introduction of the bottleneck layer decreases the performance. Despite the representation dimensionality being reduced by a factor of 50, the performance only drops by a few percent. For instance, in the three-dimensional case, the classification accuracy is still about 63%, whereas for the 512-dimensional case the accuracy increases to about 71%. Considering the benefits of having such a low-dimensional representation, such as the possibility of visualization, we consider this drop in performance acceptable. Based on these results, there is a strong indication that deep learning is indeed able to capture the relevant information of microstructure images in a very compact way. This finding has important consequences, as it implies that even with very little experimental data available, it should be possible to link the compact representation of the microstructure to properties and the processing conditions of the material. As data on properties is scarce and expensive, reducing the number of inputs for the structure–property models will lead to a better performing model[11].

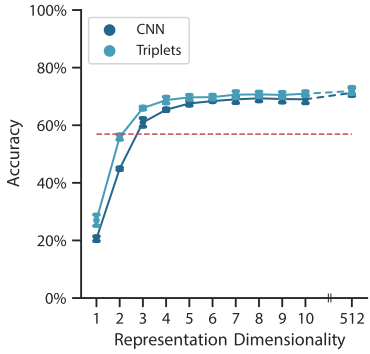
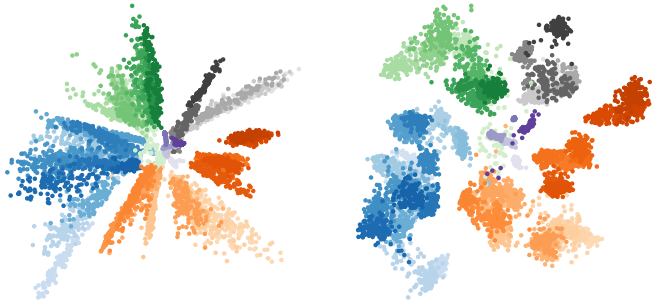


Fig. 3 The effect of the representation dimensionality on the classification accuracy. The microstructure classification accuracy of the proposed deep learning methods as a function of the representation dimension for the task of recognizing the correct material class for dataset 1. The red line indicates the best performance of the methods found in the literature. All accuracies were obtained using a random forest classifier on the microstructural representations. The error bars represent the standard deviation on three different splits of the training and test data.

In DeCost et al.[7], the t-distributed stochastic neighbor embedding dimensionality reduction technique[25] is used to create two-dimensional maps of microstructures. An advantage of the low dimensionality of our representation space is that it is possible to directly visualize the representations in a faithful way without having to rely on dimensionality techniques. Figure 4 shows the representations in two dimensions. In the left figure, we see that microstructure images belonging to the same class, represented by dots of the same colour, tend to form elongated lines. Indeed, the representations were obtained by minimizing the error on the softmax predictions, which requires the representations to be linearly separable from each other. In the figure on the right, this is not the case, as we see more compact and isolated clusters. This is because the triplet loss requires representations belonging to the same class to be as close to each other as possible. From the figures, it is noticeable that representations with similar colours tend to lay close together. Materials from the same groups have the same colourmap, which is defined in Table 1. Within each group, the intensity of the colour varies. Attention was paid to assigning similar values of intensity to microstructures that are similar in terms of processing. Especially for the triplet network, we see that the representations with lighter and darker intensities are remote from each other. This implies that the network has indeed learned a meaningful distance metric, which reflects the underlying processing of the microstructure. It is especially striking how well the different groups are separated by each other. Without any information on which materials belong to the same group, the deep learning model naturally clusters materials belonging to the same group together, indicating that the learned similarity measure corresponds well to human perception of visual similarity. It also justifies our approach of defining 60 different material classes, as within each group the representations show clear substructures that contain much more information than would have been the case if the triplet networks was trained using only the information of the material groups.



(a) (b)

Fig. 4 The representations in two dimensions. The two-dimensional representations obtained with (**a**) the bottleneck CNN and (**b**) the triplet network. The representations of the images belonging to the same class have the same colour. The colourmaps defined in Table 1 are used to indicate representations of the same group. Both models naturally learn to separate the different groups from each other. Within the group, the triplet model is also able to correctly cluster together the image representations belonging to the same material.

Table 1. Description of the different groups of materials in dataset 1.

Group id	Description	Etching	Classes	Colourmap	Accuracy
1	Austenitic matrix	Nital	1–6	Grey	81.1 ± 2.7
2	Martensitic with PAGs	Bechet–Beaujard	7–27	Orange	60.7 ± 1.6
3	Tempered martensitic	Nital	28–46	Blue	60.5 ± 1.1
4	Quenched martensitic	Nital	47–55	Green	61.6 ± 5.9
5	Ferritic steels	Nital	56–60	Purple	97.6 ± 2.3

Dataset 1 only contains martensitic, austenitic and ferritic microstructures. To each group, we have assigned a specific colourmap, which is used in the plots in the following sections. We also mention the etching procedures that were used to obtain the images, as it can greatly impact the visual appearance of the microstructure. We also list the corresponding labels of the material classes that belong to these groups. The last column lists the classification accuracies within the groups of materials for the three-dimensional triplet representation using a random forest classifier.

Comparing the network performance to human experts

As deep learning models extract information from the pixel level, they have the potential to extract much more complex patterns than would be possible for humans. From the results in the previous section, we already know that it is possible to correctly recognize more than 60% of the crops. This suggests that the model is capable of distinguishing between materials with only small differences in processing conditions, which would be indistinguishable for the human eye, and triggers further investigation. On the other hand, 60% is well below typical values reported in literature, which might be an indication that some material classes cannot be discerned from each other.

Figure 5 shows the confusion matrix for microstructure classification on the test set, using random forests applied to the three-dimensional triplet representations. All results are averaged over three independent splits of training and test set. We have indicated the groups of the materials, as defined in Table 1, with coloured squares. As was already noted in Fig. 4, the model nearly perfectly assigns each image to the right group. We find that 98.9% of the crops are correctly assigned. This is a high score and likely on par with what a human expert could achieve. Even for experts obtaining 100% accuracy would be difficult, as it is for instance hard to tell the difference between quenched (group 3) and tempered martensite (group 4) based on a single 200 × 200 crop.

The accuracy for the different groups is listed in Table 1. We see that within each group the accuracy is still above 50%. This is remarkable, as the materials within groups are visually very similar. A small sample of experts saw no clear difference between the materials belonging to the same group based on a single 200 × 200 image. There are big discrepancies in the accuracies of the different groups. For groups 1 and 5, the accuracy is significantly higher than for the other groups. We identify two reasons why this might be the case. First, these group contain fewer materials than the other groups, making it a priori easier to correctly recognize the right material. Second, the compositional ranges of these two groups, which are given in the Supplementary Table 1, are much wider than for the other groups. Groups 2 and 3 have a comparable number of classes and similar compositional ranges. Their performance is similar, but the Nital etching used in group 3 seems to reveal more relevant features for the machine learning algorithm than the Bechet–Beaujard etching used for group 2. For group 4, we note that the variance among the different runs is substantially higher than that of the other groups. We find that the performance of this group is most sensitive to the images that are used as a test set. In some cases, the test set mainly consists of images that are taken closer to the edge of the material, whereas there are only few such examples in the training set. The location where the images are taken seems to greatly affect the model performance and explains why, despite the group containing only eight materials, we obtain a relatively low accuracy for this group. More metrics such as precision and recall for both recognition tasks can be found in the Supplementary Tables 10 and 11. Lastly, it is clear from Fig. 5 that there are some material classes between which the model often confuses, especially for groups two, three

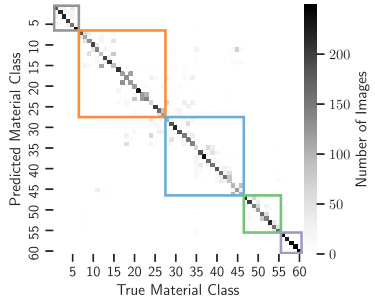


Fig. 5 The confusion matrix of the three-dimensional triplet model for the first dataset. The greyscale value indicates the number of images that belong to the material class shown on the *x*-axis and are assigned by the model to the material class shown on the *y*-axis. For the ideal model, only the diagonal would be coloured. The coloured boxes show the groups using the colourmaps that are defined in Table 1. The model mainly confuses materials belonging to the same group, which are visually very similar.

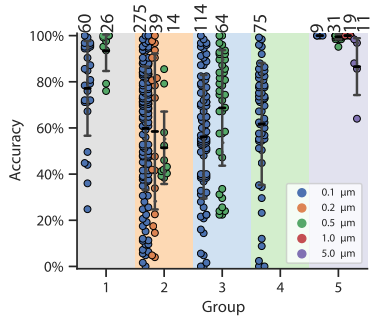


Fig. 6 The performance of the three-dimensional triplet network by group and magnification. The model is able to deal with different magnifications, measured by the inter-pixel distance, without being explicitly taught to do so. The number on top of each column indicates the number of images in the dataset for the given group and magnification. The background colouring uses the colourmaps defined in Table 1. The thick black lines represent the mean accuracy and the error bars represent the Standard Deviation (SD) on three different splits of the training and test data. The dots depict the average accuracy of all crops belonging to the same image.

and four. It is an indication that in these groups some of the material classes are indistinguishable. In Supplementary Fig. 5, we study the effect of the number of classes more explicitly. We show that it is possible to obtain more than 90% accuracy, while retaining more than 30 different material classes.

Analysing the effect of different magnifications

It is interesting to analyse how our method deals with the different amount of training data for the different magnifications. As can be seen in the Supplementary Fig. 2, the dataset contains images at different magnifications for many of the materials, but the number of images for each magnification differs strongly. From Fig. 6, we find that the model achieves a better score on the lower magnifications (0.5 µm inter-pixel distance) for the first group, even though there are more than twice as many training images for the lower magnifications (0.1 µm inter-pixel distance). For group 2, we see an opposite trend, as the accuracy decreases with decreasing magnification. Especially for the 0.5 µm magnification, the performance drops significantly. This is because sufficient detail is no longer present at this magnification to correctly classify the images, as is confirmed by human experts. For group 3, we observe a similar trend as for group 1 with lower magnifications being preferred, even tough fewer images are present in the dataset. For group 5, all magnifications have very high scores, except for the 5.0 µm magnification, which no longer contains sufficient detail about the material. There are two main observations from this all. First, the model is able to cope very well with different magnifications. A possible explanation is that the grouped convolutions in the ResNeXt architecture allow the model to look at the image at different length scales at the same time. A second observation is that the model has a clear preference for certain magnifications, and that this preference strongly depends on the material group. This can be understood by considering the trade-off between statistical representativity, which requires the image to cover a sufficiently large surface of the material, and sufficient detail, which requires the image to have a sufficiently high resolution. As the model examines the images at the pixel level, it is able to detect small details in the image very well, but it requires to examine a larger surface of the material to correctly recognize the microstructure. This can also be seen in the examples in Fig. 7. The yellow regions for groups one, two and five are areas where the model doubts between several materials, because it cannot detect enough relevant features to correctly recognize the material based on a single crop. At lower magnifications, crops will contain more relevant information. For groups three and four, the opposite is the case and the model recognizes clear features from two different classes in the yellow regions. This implies that for these two groups the 200×200 crops already contain a lot of information. We conclude that the model will prefer lower magnifications provided there is still enough detail in the image and that the preferred magnification is strongly dependent on the material under consideration.

We have also included the average accuracy per image in Fig. 6. We find that analysing such averages is helpful in detecting where the model systematically fails to correctly recognize the correct material class. In the Supplementary Figs. 9–13, we show some example images that are completely misclassified. It turns out that although the model is relatively robust to irregularities such as dust particles on the microscopy, regions that are slightly out of focus and the presence of over-etched regions, it is sensitive to the precise etching procedure and this can overlay the subtle differences between some of the material classes.

Interpreting the model predictions

To shed more light on where the model looks at, we propose a simple method based on averaging predictions of many crops belonging to the same image. Per pixel, we average the predictions of all crops that contain that specific pixel. Thus, we obtain a probability distribution per pixel over the different material classes. In Fig. 7, we show the result of this procedure for a typical image of each group, which was taken from the test set. Based on these heatmaps, we obtain a clear idea of where the model looks at to recognize the materials. However, the reason why specific regions are more easily recognized is not always immediately clear. We briefly discuss an example for each group and give a possible explanation. For the first group, we see that the model mainly looks at the density of precipitates and of so-called pearlitic lamellae to assign the image to the right material. Also, the presence of triple junctions such as the one in the upper centre of the image seems to affect the model predictions. For the second group, we see that mainly the grain size affects the model decision. The regions with larger grains are assigned to the right material, whereas the regions with smaller grains are assigned to

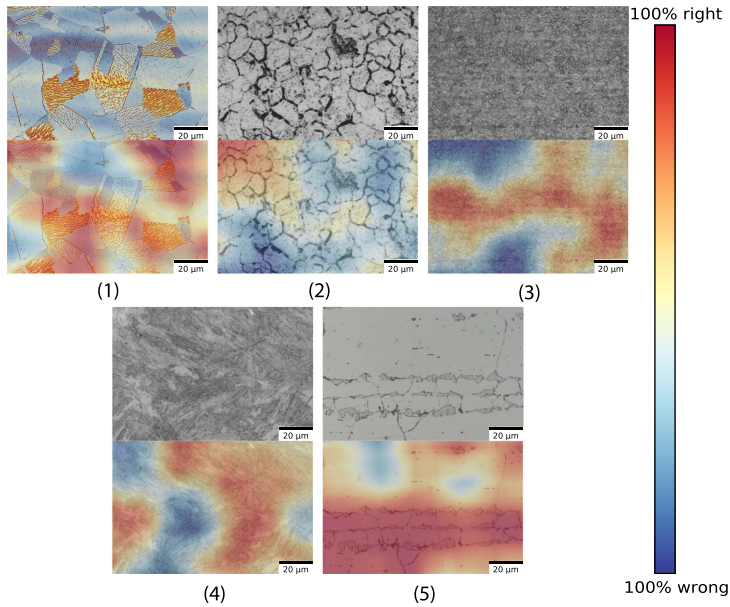


Fig. 7 Heatmaps of the three-dimensional triplet network predictions. By averaging the predictions of 10,000 crops taken randomly from the image, we can construct heatmaps of regions where the model looks at for assigning an image to a material class. For each of the material groups, one of the original images of the test set is shown on top and the overlaid heatmap is shown below. The colourmap is chosen such that a region on the image is marked red if the model assigns this region to the right material and blue if the region is assigned to a different material. Yellow indicates that model is unsure of which class the material should be assigned to.

other materials. Regions with medium sized grains are marked yellow, as the model doubts between the right material and other materials. In the third group, we show an image where the model fails to correctly classify the material. The presence of elongated prior austenitic grains causes the model to assign this image to a different material class. In the regions where these grains are rounder, the model doubts between the right material class and the other. A closer inspection of the training data, shows that the prior austenitic grains are typically indeed rounder for the correct material class. Hence, it is understandable why the model is confused. In the fourth group, we see some clear regions that are marked in blue. The blue region in the centre of the image has darker, elongated grains, whereas the blue region in the upper left corner has rounder grains. Although these two regions are marked blue, they are very different and further analysis shows that these two regions are indeed assigned to different material classes by the model. In the image of the last group, we see that areas with small, elongated grains and second phases (the small polygonal areas) are marked blue. This combination of small, elongated grains and second phases is not the most representative of this class and similar features are seen in the classes the model confuses with . Our analysis indicates that the model has learnt that physically relevant features such as the size and shape of grains are important to distinguish between different materials. A larger version of the images can be found in the Supplementary Figs. 14–18.

Comparing the triplet representations to other representations

We benchmark the discriminative power of the presented method to other microstructural representations found in the literature by evaluating the accuracy on two different microstructure recognition tasks. The first task is to correctly recognize the group, as defined in Table 1, to which the microstructure on the image crop belongs. The second task is to correctly recognize the specific material. More information on the implementation of the other microstructural representations can be found in the Supplementary Methods.

In Table 2, we show the test accuracies for both microstructure recognition tasks. We see that the task of recognizing the right group results in very high accuracies, in accordance with what is reported in the literature. As expected, the recognition of the right material turns out to be much harder. It is noteworthy that the older and more conventional microstructural representations such as the correlations, keypoints and morphological features do not have a very high discriminative power. These features were handcrafted by metallurgists to distinguish different metallurgical phases, similar to the groups we are considering and were not designed to spot small differences in microstructures. Texture-based methods such as the Haralick features and the texture features introduced in Webel et al.[26] achieve very good accuracies despite their low dimensionality. Still, we find that mainly deep learning-based features obtain the highest accuracies. Of all methods found in the literature, we find that the VGG16 C_{43} features with mean pooling obtain the best performance, which is in line with the findings in DeCost et al.[7]. However, this representation has a dimensionality of 512, which is much higher than what we aim for in this paper. We see that the triplet networks presented in this paper outperform the other methods found in the literature. This is not unexpected, as the network used to create these features was specifically trained on the dataset, whereas other methods use the network output of a standard pretrained model without further optimization of the

Table 2. Comparison of the classification accuracy of several microstructural representations on dataset 1.

Method	Dimensionality	Test acc. group [%]	Test acc. material [%]
Haralick[5]	13	94.4 ± 0.2	46.5 ± 2.3
lbp[5]	20	94.3 ± 0.3	34.4 ± 1.1
Greyco[5]	4	72.1 ± 0.5	18.5 ± 0.5
Correlations[8]	20	78.0 ± 1.0	19.7 ± 1.9
Surf[15]	100	69.1 ± 0.9	13.6 ± 0.8
VGG16 VLAD C_{43}[7]	16,384	62.7 ± 6.5	48.3 ± 6.6
VGG16 mean C_{43}[7]	512	99.2 ± 0.3	56.2 ± 1.5
VGG16 max C_{43}[7]	512	97.9 ± 0.3	47.6 ± 2.3
Morphological[6]	21	77.1 ± 2.4	17.5 ± 1.0
Texture[26]	8	90.6 ± 1.5	44.4 ± 3.8
Morph. + texture[6]	29	93.4 ± 1.0	45.8 ± 3.8
Morph. + texture + genetic algorithm[12]	16	93.6 ± 0.0	47.8 ± 0.0
Triplets	2	98.8 ± 1.8	55.6 ± 1.1
Triplets	3	99.1 ± 1.8	65.9 ± 0.7
Triplets	10	99.2 ± 1.5	71.0 ± 1.2

We show the accuracy (acc.), both for the task of correctly recognizing the right group (see Table 1) and for recognizing the right material. All accuracies were obtained by using a random forest classifier on crops coming from images of the test set. We also list the SDs, which are computed by repeating the entire training procedure three times with different splits for the train and test data. The proposed models clearly perform better than other models found in the literature, while using much more compact representations.

Table 3. Comparison of the classification accuracy of several microstructural representations on dataset 2.

Method	Dimensionality	Out-of-bag acc. [%]	Test acc. [%]
Haralick[5]	13	98.4 ± 0.3	91.5 ± 3.2
lbp[5]	20	92.0 ± 1.0	83.6 ± 4.2
Greyco[5]	4	63.0 ± 1.0	50.9 ± 1.0
Correlations[8]	20	79.8 ± 1.0	69.8 ± 2.6
Surf[15]	100	28.1 ± 1.4	28.4 ± 1.2
VGG16 VLAD C_{43}[7]	16,384	65.5 ± 3.8	77.3 ± 0.8
VGG16 mean C_{43}[7]	512	99.4 ± 0.1	97.8 ± 0.5
VGG16 max C_{43}[7]	512	98.6 ± 0.2	96.1 ± 2.2
Morphological[6]	21	72.0 ± 1.6	66.7 ± 0.7
Texture[26]	8	89.6 ± 1.9	83.4 ± 2.2
Morph. + texture[6]	29	93.3 ± 1.1	88.9 ± 1.8
Morph. + texture + ga[12]	16	92.7 ± 1.9	87.5 ± 1.8
Triplets	2	71.7 ± 4.8	66.2 ± 6.5
Triplets	3	83.7 ± 3.9	78.9 ± 5.3
Triplets	10	97.7 ± 0.6	94.6 ± 2.0
Triplets	512	99.9 ± 0.1	99.5 ± 0.4

Both the accuracy (acc.) for the test data and the out-of-bag accuracy are listed. All accuracies were obtained using a random forest classifier. We also list the SDs, which are computed by using threefold cross-validation. Even on unseen materials the proposed methods perform as well as the other representations with similar dimensionality.

model parameters. Still, our method outperforms other methods with such small feature vectors. Already with a two-dimensional representation, we can get a performance comparable to the best methods found in the literature. These results suggest that the currently used methods yield vector representations that are much larger than they should be.

Supplementary Table 12 gives more results, where some other less performing methods are included. We also list the out-of-bag accuracy of the random forest on the training set, which should be a good indication of the performance on the test set[24]. The out-of-bag accuracy is however significantly higher than the accuracy on the test set for all features. This is due to the fact that the images used for training the decision trees and the ones left out for determining the out-of-bag score overlap, as they are crops from the same original images. This stresses the need of examining new independent images to have an unbiased evaluation of the model performance. In our comparison, we systematically use a crop size of 200×200, because most modern deep learning architectures are trained on crops of similar size. Therefore, the crop size might favour deep learning method compared to ways of representing the microstructure. In the Supplementary Fig. 7, we study the effect for the crop size in more detail for the Haralick features. We find that using larger crop sizes can increase the performance by another 3%.

Assessing the generalization to new materials

As the dataset, which we used to train our models, contains five different groups of materials, it is interesting to check how well the representations generalize to unseen types of materials. In the literature, this is often referred to as zero-data learning or zero-shot learning[27,28]. To this end, we use the triplet network that was trained on dataset 1 to obtain representations for the images of dataset 2. These representations are then again used to train a random forest classifier to recognize the ten predefined material classes of dataset 2. Thus, we can assess how well the representations that are computed by the triplet network generalize to new groups of materials. As before, we make a comparison to other methods found in the literature.

To measure the performance of each of the representations, we show in Table 3 the out-of-bag accuracy on the training set and the test accuracy. We use threefold cross-validation and for each fold we train the model on crops coming from two images of each class, and we use the crops of the third image as a test set. Thus, we never use crops from the same image simultaneously in the training and test set.

Many representations obtain accuracies close to 90%. This is expected, as there are clear visual differences between most of the material classes, which was not the case for the materials in dataset 1. There is a clear tendency for higher dimensional representations to perform better. The only exceptions are the VLAD-encoded representation, of which the dimensionality is too high to learn meaningful patterns for the small dataset under consideration. The triplet network representations perform at least as good as other representations of similar dimensionality, which indicates that they generalize well to new materials. However, the difference in performance between the three- and ten-dimensional representations is much bigger than for dataset 1. This seems plausible, as trying to capture all relevant information of a dataset in only two or three dimensions requires the model to learn features that are more specific to the materials in the training set compared to the ten-dimensional case.

In Fig. 8, we show the confusion matrix of the representations from the three-dimensional triplet network. The model mainly confuses between crops from material classes 4, 6 and 8. These classes are indeed visually similar, as they all have features from both bainitic and quasi-polygonal ferritic steel. Furthermore, bainite was not included in the training set, so that we would

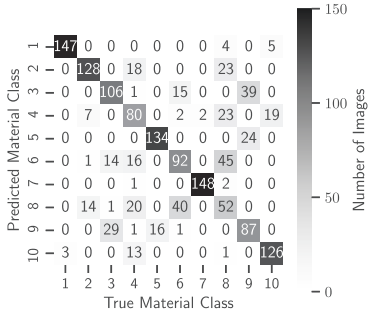


Fig. 8 The confusion matrix of the three-dimensional triplet model for the second dataset. The model obtains a decent performance for all materials, but struggles most with materials that were not included in the original training set.

indeed expect a lower performance compared to the other material classes.The model deals relatively well with all materials and, as before, we find that the triplet representations are able to quantitatively express visual similarity.

It is clear that there is an unavoidable trade-off between the dimensionality of the microstructural representation and its generalizability. Low-dimensional representations tend to be either unable to capture enough detail or are too much tailored to the materials in the training data. As was the case for recognizing the correct group in dataset 1, we obtain near-perfect accuracies on this dataset. This is once more an indication that microstructure recognition of microstructures with clear visual differences is no longer a challenging topic. Research should instead focus on datasets with visually similar materials and possibly try to go beyond the capabilities of human experts.

We propose triplet networks as a method capable of learning highly compact representations of microstructure images. We demonstrate that by using information on the composition and processing, it is possible to obtain very detailed microstructural representations. Despite being low-dimensional, these representations contain sufficient information to discern visually similar materials that only have small differences in composition and processing conditions. Capturing these small differences is a prerequisite for structure–property prediction, as small variations in composition and processing can greatly affect the properties. Furthermore, the method is able to cope well with a large range of magnifications. We introduce a visual way of interpreting the representations and find that they behave surprisingly similar to how expert metallurgists analyse microstructure images, focusing on features such as grain edges and precipitate densities. We present a comparative benchmark of different microstructural representations found in the literature for a microstructure recognition task. Already in two dimensions, the triplet network representations obtain a performance that is competitive with the state-of-the-art. When applying the representations to new materials, the generalization performs at least as good as other representations with the same dimensionality. We observe a clear trade-off between compactness and generalization, as high-dimensional representations perform significantly better than the more low-dimensional ones. As triplet networks learn how to represent a microstructure directly from the data, the representations should become even better when trained on a larger dataset of microstructure images. As the size of microstructure image datasets is rapidly increasing, we expect the presented method has a high potential as a tool for microstructure analysis.

METHODS

Obtaining the training and test set

In order to correctly evaluate the model performance, it is important to split the dataset in a training and test set, so that both are representative and independent. The training set is used to train the model, while the test set is only used to evaluate the model performance. In the first dataset, we use for each class 80% of the images as the training set. Each steel is considered to be of a different class when the processing conditions and composition of the steel differ, regardless of whether the properties of the steels differ or whether it is possible to visually distinguish between the materials. Although this is an objective criterion to define different materials classes, there is no a priori guarantee that it is possible to recognize the right material purely based on an optical microscopy image. Still, by introducing more material classes than is usually done, we aim to let the triplet network learn very detailed representations of the microstructure. Once we have these detailed representations, we apply them to recognize both the correct material class and the correct material group. As the dataset is highly unbalanced in terms of microstructure images per class, we randomly select crops of the original images with a 200×200 pixel resolution until we have 500—possibly overlapping—images in each class. A test set is created by taking, for each class, the remaining 20% of the original images with a 1000×1200 pixel resolution that are not included in the training set and by randomly selecting a total of 125 crops with 200×200 resolution from these images. The training and test set will hence never contain crops from the same image. We repeat the outlined procedure to create crops for the training and test set three times. All reported results are the average performance over these three different train and test splits, and where relevant we also mention the SD over these three splits.

For the second dataset, 50 randomly sampled 200×200 crops were taken from each image. Threefold cross-validation is used to obtain a reliable assessment of the model performance.

Training the triplet network

For our starting model, we adopt a ResNeXt50 convolutional architecture[19], which has over 25 million trainable parameters and was pretrained on the ImageNet dataset[29]. We choose this architecture, because it performed better than other architectures with similar complexity in our experiments, which is discussed in more detail in the Supplementary Table 3. We use pretrained weights only for the convolution layers of the model. The fully connected layers at the end of the network are trained from scratch and we are therefore free to choose the architecture of these layers. All models are implemented using the PyTorch deep learning framework[30] and fine-tuning of the pretrained model is performed using FastAI[31].

Data augmentation is used to artificially increase the number of images in the training set. We randomly apply Gaussian blur, rotations and changes in lightening to the crops to make the model invariant to changes in these conditions. Details on the exact augmentations used in this work can be found in Supplementary Table 6.

One of the challenges of training a triplet network is the selection of the triplets. As it is computationally unfeasible to select all possible triplets, a selection has to be made. Although various criteria have been used in the literature[32–34], we use the batch semi-hard criterion, as it prevents the representations from collapsing onto the same point[23], which was a problem for the other criteria. For the training of the triplet network, we use the bottleneck models discussed in the previous section as starting point and retrained the dense layers at the end of the network, keeping the other parameters fixed. This helps to reduce the number of trainable parameters and thus the memory usage. The details of the hyperparameters used for training the CNN model and the triplet network can be found in the Supplementary Tables 4 and 5.

Choosing the right classification technique

All accuracies reported in this work are obtained by training a random forest model[24]. There are two main reasons to prefer this classifier over other techniques. First, we found that it does not require an additional validation set to tune the hyperparameters of the model if one starts from reasonable default values, thus eliminating the need for additional data. This finding is supported in the literature[35]. Furthermore, the generalization performance of a random forest can also be assessed by looking at the out-of-bag accuracy on the training set[24]. The out-of-bag accuracy relies on the fact that each individual decision tree of the random forest is trained on a subset of the training data. It is obtained by evaluating the

model performance of each sample on the trees that did not use it for training. Second, we found that it yields good results both in low- and high-dimensional feature spaces, allowing for a more objective comparison of features of different dimensionality. Support Vector Machines[36] are another commonly used method for microstructure recognition[6,12,15]. However, we found that such models tend to perform worse in high-dimensional spaces and are very sensitive to the tuning of the hyperparameters[37]. The details of the used model hyperparameters can be found in the Supplementary Table 7.

DATA AVAILABILITY

The data that support the findings of this study are available in www.microstructuredb.com/papers. The weights of some of the models are available on www.microstructuredb.com/papers.

CODE AVAILABILITY

The code that supports the findings of this study is available on www.microstructuredb.com/papers.

Received: 1 October 2019; Accepted: 23 September 2020;
Published online: 15 October 2020

REFERENCES

1. Zaefferer, S., Ohlert, J. & Bleck, W. A study of microstructure, transformation mechanisms and correlation between microstructure and mechanical properties of a low alloyed TRIP steel. *Acta Mater.* **52**, 2765–2778 (2004).
2. Olson, G. B. Computational design of hierarchically structured materials. *Science* **277**, 1237–1242 (1997).
3. Panchal, J. H., Kalidindi, S. R. & McDowell, D. L. Key computational modeling issues in Integrated Computational Materials Engineering. *Comput. Aided Des.* **45**, 4–25 (2013).
4. Torquato, S. Statistical description of microstructures. *Annu. Rev. Mater. Res.* **32**, 77–111 (2002).
5. Chowdhury, A., Kautz, E., Yener, B. & Lewis, D. Image driven machine learning methods for microstructure recognition. *Comput. Mater. Sci.* **123**, 176–187 (2016).
6. Gola, J. et al. Advanced microstructure classification by data mining methods. *Comput. Mater. Sci.* **148**, 324–335 (2018).
7. DeCost, B. L., Francis, T. & Holm, E. A. Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. *Acta Mater.* **133**, 30–40 (2017).
8. Fullwood, D. T., Niezgoda, S. R. & Kalidindi, S. R. Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. *Acta Mater.* **56**, 942–948 (2008).
9. Schmidhuber, J. Deep learning in neural networks: an overview. *Neural Networks* **61**, 85–117 (2015).
10. Hoffer, E. & Ailon, N. In *International Workshop on Similarity-Based Pattern Recognition* 84–92 (Springer, Cham, 2015).
11. Friedman, J., Hastie, T. & Tibshirani, R. *The Elements of Statistical Learning* (Springer, New York, 2001).
12. Gola, J. et al. Objective microstructure classification by support vector machine (SVM) using a combination of morphological parameters and textural features for low carbon steels. *Computat. Mater. Sci.* **160**, 186–196 (2019).
13. Azimi, S. M., Britz, D., Engstler, M., Fritz, M. & M ücklich, F. Advanced steel microstructural classification by deep learning methods. *Sci. Rep.* **8**, 1–14 (2018).
14. Olson, G. B. & Azrin, M. Transformation behavior of TRIP steels. *Metall. Trans. A* **9**, 713–721 (1978).
15. DeCost, B. L. & Holm, E. A. A computer vision approach for automated analysis and classification of microstructural image data. *Comput. Mater. Sci.* **110**, 126–133 (2015).
16. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at https://arxiv.org/abs/1409.1556 (2014).
17. DeCost, B. L., Lei, B., Francis, T. & Holm, E. A. High throughput quantitative metallography for complex microstructures using deep learning: a case study in ultrahigh carbon steel. *Microsc. Microanal.* **25**, 21–29 (2019).
18. Ajioka, F., Wang, Z.-L., Ogawa, T. & Adachi, Y. Development of high accuracy segmentation model for mmicrostructure of steel by deep learning. *ISIJ Int.* **60**, 954–959 (2020).
19. Xie, S., Girshick, R., Dollar, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Computer Vision and Pattern Recognition* 5987-5995 (IEEE, New York, 2017).
20. Baldi, P. In *ICML Workshop on Unsupervised and Transfer Learning* 37–49 (Microtome, Brooklyn, 2012).
21. Pratt, L. Y. In *Advances in Neural Information Processing Systems* 204–211 (Morgan Kaufmann, Burlington, 1993).
22. Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. In *Advances in Neural Information Processing Systems* 3320–3328 (Curran, Brooklyn, 2014).
23. Schroff, F., Kalenichenko, D. & Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conf. Computer Vision and Pattern Recognition* 815–823 (IEEE, New York, 2015).
24. Breiman, L. Random forests. *Machine Learn.* **45**, 5–32 (2001).
25. Maaten, L. V. D. & Hinton, G. Visualizing data using t-SNE. *J. Machine Learn. Res.* **9**, 2579–2605 (2008).
26. Webel, J., Gola, J., Britz, D. & Mücklich, F. A new analysis approach based on Haralick texture features for the characterization of microstructure on the example of low-alloy steels. *Mater. Charact.* **144**, 584–596 (2018).
27. Larochelle, H., Erhan, D. & Bengio, Y. Zero-data learning of new tasks. In *AAAI Conf. Artificial Intelligence* 646–651 (AAAI, Palo Alto, 2008).
28. Rohrbach, M., Stark, M. & Schiele, B. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *IEEE Conf. Computer Vision and Pattern Recognition* 1641–1648 (IEEE, New York, 2011).
29. Smith, J. et al. Linking process, structure, property, and performance for metal-based Additive Manufacturing: computational approaches with experimental support. *Comput. Mech.* **57**, 583–610 (2016).
30. Paszke, A. et al. In *Advances in Neural Information Processing Systems* 8026–8037 (Curran, Brooklyn, 2019).
31. Howard, J. & Gugger, S. Fastai: a layered API for deep learning. *Information* **11**, 108 (2020).
32. Wu, C.-Y., Manmatha, R., Smola, A. J. & Krahenbuhl, P. Sampling matters in deep embedding learning. In *The IEEE Int. Conf. Computer Vision* 2859–2867 (IEEE, New York, 2017).
33. Oh Song, H., Xiang, Y., Jegelka, S. & Savarese, S. Deep metric learning via lifted structured feature embedding. In *The IEEE Conf. Computer Vision and Pattern Recognition* 4004–4012 (IEEE, New York, 2016).
34. Hermans, A., Beyer, L. & Leibe, B. In defense of the triplet loss for person re-identification. Preprint at https://arxiv.org/abs/1703.07737 (2017).
35. Probst, P., Wright, M. N. & Boulesteix, A.-L. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining Knowledge Discov.* **9**, e1301 (2019).
36. Cortes, C. & Vapnik, V. Support-vector networks. *Machine Learn.* **20**, 273–297 (1995).
37. Duarte, E. & Wainer, J. Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters. *Pattern Recognit. Lett.* **88**, 6–11 (2017).

ACKNOWLEDGEMENTS

M.L. and S.C. acknowledge financial support from OCAS NV by an OCAS-sponsored PhD position and by an OCAS-endowed chair at Ghent University, respectively. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation–Flanders (FWO) and the Flemish Government–department EWI.

AUTHOR CONTRIBUTIONS

M.L. contributed to the methodology, software, formal analysis and writing of the original draft. M.S. contributed to the methodology and writing of the original draft. K.T and L.D. provided the datasets and contributed to the writing and to the formal analysis. T.D. was responsible for supervision and funding acquisition. S.C. was responsible for supervision, writing and funding acquisition. All authors reviewed the final manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

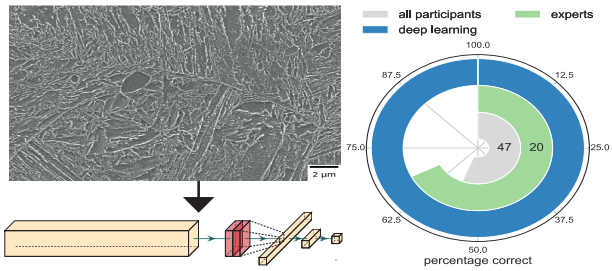
Supplementary information is available for this paper at https://doi.org/10.1038/s41524-020-00423-2.

Correspondence and requests for materials should be addressed to M.L.

Reprints and permission information is available at http://www.nature.com/reprints

Paper II

Race against the Machine: can deep learning recognize microstructures as well as the trained human eye?



M. Larmuseau, M. Sluydts, K. Theuwissen,
L. Duprez, T. Dhaene and S. Cottenier

Scripta Materialia, **2020**, *193*, 33–37

M. Larmuseau conceived the methodology, wrote the software, performed the calculations, interpreted the results and prepared the manuscript.

Scripta Materialia 193 (2020) 33–37

Contents lists available at ScienceDirect

Scripta Materialia

journal homepage: www.elsevier.com/locate/scriptamat

Race against the Machine: can deep learning recognize microstructures as well as the trained human eye?

Michiel Larmuseau [a,b,c,*], Michael Sluydts [a,b], Koenraad Theuwissen [d], Lode Duprez [d], Tom Dhaene [c], Stefaan Cottenier [a,b]

[a] Center for Molecular Modeling, Ghent University, Technologiepark 46, Zwijnaarde, B-9052, Belgium
[b] Department of Electromechanical, Ghent University, Technologiepark 46, Zwijnaarde, B-9052, Belgium
[c] IDLab, Department of Information Technology, Ghent University – IMEC, Technologiepark 126, Zwijnaarde, B-9052, Belgium
[d] OCAS NV/ArcelorMittal Global R&D Gent, Pres. J. F. Kennedylaan 3, Zelzate, B-9060, Belgium

ARTICLE INFO

Article history:
Received 23 July 2020
Revised 8 October 2020
Accepted 12 October 2020

Keywords:
Image analysis
Steels
Modeling
Scanning electron microscopy (SEM)

ABSTRACT

The promising results of deep learning in image recognition suggest a huge potential for microscopic analyses in materials science. One major challenge for its adoption in the study of materials is the limited number of images that are available to train models on. Herein, we present a methodology to create accurate image recognition models with small datasets. By explicitly taking into account the magnification and by introducing appropriate transformations, we incorporate as many insights from material science in the model as possible. This allows for a highly data-efficient training of complex deep learning models. Our results indicate that a model trained with the presented methodology is able to outperform human experts.

Microscopy images are an important source of information on the small-scale structure of materials, referred to as the microstructure. However, most of the time, this information is analysed only qualitatively: domain experts for instance mainly examine these images to assess whether the processing of the material went well. While such an assessment is important, a lot of information contained in the microscopy image is not used. In recent years, approaches towards a more quantitative analysis of these images using machine learning have been thoroughly investigated in literature[1–4]. However, all these approaches rely on generic microstructure descriptors that are not tailored to the specific materials in the dataset, resulting in sub-optimal performance.

Deep learning methods[5] make it possible to learn microstructural descriptors directly from the available data. Despite the first report of deep learning outperforming humans in an image recognition task already dating back to 2011[6], its adoption in practical material science remains limited[7–9]. A possible explanation for this, is that deep learning is often only deemed to outperform classical methods when large datasets of images are available. With most commonly used datasets in material science literature containing around the order of a thousand images[2,4,10], this opinion feels reasonable. However, recent work[11] has shown that by creating tailor-made deep learning models for specific datasets, one can outperform models that use generic microstructure descriptors.

This would make it possible to investigate how well deep learning models perform in recognizing microstructures compared to expert materials scientists. Although it is clear from other fields in computer vision that deep learning models can outperform human experts[12,13] provided a sufficient number of images, we here aim to investigate the performance of models that are trained on only around a hundred microstructure images in total. A hundred images is a practical amount, as it can easily be collected in a systematic study of a class of materials.

To evaluate the performance of both a neural network that is obtained using the methodology presented in [11] and the panel of experts, we have organized two different quizzes. For the first quiz, 36 microscopy images need to be assigned to one of the five pre-defined classes of microstructures. The option “None of these” is included in case a microstructure image is shown that does not belong to one of the five pre-defined classes. The dataset on which the model was trained, the training set, contains 134 images in total, with magnifications, expressed as pixels per micrometer, ranging from 1.1 to 212 pixels per micrometer ($pp\mu m$). Both optical microscopy images and scanning electron microscopy (SEM) images have been included. An example for each type of microstructure is shown in Fig. 1 (a). For the second quiz, 21 SEM microscopy images of complex martensitic steels need to be assigned to one

* Corresponding author.
E-mail address: michiel.larmuseau@ugent.be (M. Larmuseau).

https://doi.org/10.1016/j.scriptamat.2020.10.026
1359-6462/

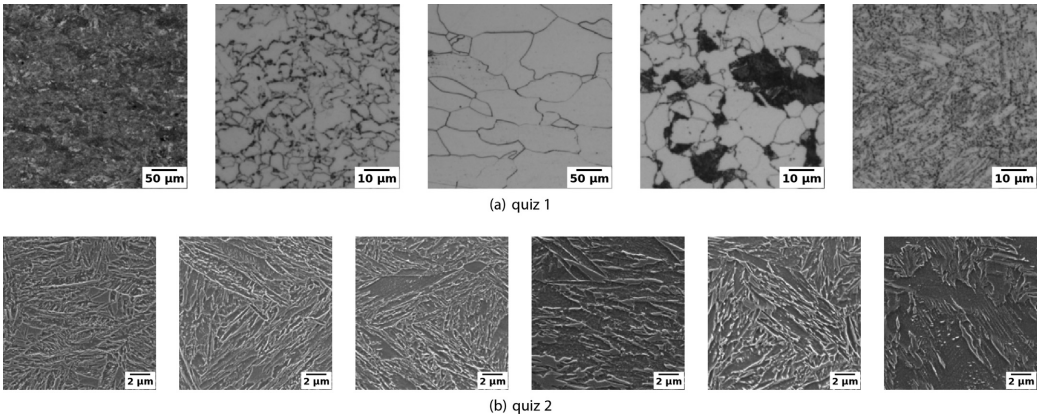


Fig. 1. An illustration of the microstructures for each of the pre-defined material classes for quiz 1 (a) and quiz 2 (b). More information on the material classes can be found in Supplementary S1 and S2.

of the six pre-defined classes of microstructures. Here, the option "None of these" is not included for the panel of experts, mainly to reduce the difficulty of the quiz. The training set contains 58 images in total, with magnifications of either $53\,pp\mu m$ or $106\,pp\mu m$. Some examples of these images are shown in Fig. 1 (b). In Supplementary S1 and S2, we give a description of each of the material classes and show the distribution of the magnifications for each microstructure type for both quizzes.

Since we want to train a model with only a few images of each microstructure class, it is necessary to include as many insights from material science as possible. For instance, the model should be robust to imaging conditions such as lighting, etching and dust particles. We achieve this by artificially modifying the images during training and in this way augmenting the information shown to the model. Concretely, we have made use of the following modifications:

Aggressive data augmentation transforms the images before they are used for training the model. These transformations include generic ones such as rotations, mirroring and changes in brightness. Additional transformations were conceived to better highlight the relevant features in a microscopy image. Local blurring of the image is used to mimic for instance a dust particle covering a certain part of the image. Edge detectors stress the importance of phase boundaries. Warping of the image explicitly includes the translational invariance of the image in combination with other transformations.

Crops at different length scales are used to further increase the number of images on which the model is trained. Given an original image, which has a pixel resolution of 1000 × 1200, a smaller crop of the image is randomly selected and either down- or up-sampled, so that the final crop used for training has a resolution of 200 × 200 pixels. Because of this, the model can learn to recognize the material at different length scales.

Explicit inclusion of the magnification serves as an additional input to the model. For each randomly selected crop, the number of dots per micrometer is computed and used as input for the deep learning model. This is deemed necessary, as in microscopy images there can be a large range of magnifications compared to classification problems in other fields of computer vision. For instance, in the training set for quiz 1 there is a factor of 265 difference between the smallest and the largest magnification. Note that this in fact mimics how human experts classify images, as they would always require a scale-bar in order to classify a material.

The machine learning approach used in this work consists of two steps: 1) obtaining the microstructural descriptors and 2) assigning these descriptors to a material class. In the first step, we train a deep neural network[5] to learn descriptors of a microstructure image through the use of triplet networks[14]. The neural network takes as input both a transformed crop $\mathbf{x_i}$ and its resolution l_i, expressed as dots per micrometer, and outputs a vector $\mathbf{f}(\mathbf{x_i}, l_i)$. We will refer to the elements of this vector as the descriptors of the microstructure image. The deep neural network has several millions of parameters, which are iteratively updated by minimizing for each crop $\mathbf{x_i^a}$ the triplet loss[15]:

$$\max\left(0, ||\mathbf{f}(\mathbf{x_i^a}, l_i^a) - \mathbf{f}(\mathbf{x_i^p}, l_i^p)|| - ||\mathbf{f}(\mathbf{x_i^a}, l_i^a) - \mathbf{f}(\mathbf{x_i^n}, l_i^n)|| + \alpha\right), \quad (1)$$

where $||.||$ represents the euclidean norm, $\mathbf{x_i^a}$ and $\mathbf{x_i^p}$ are crops belonging to the same material class and $\mathbf{x_i^n}$ is a crop belonging to a different material class. The α is a positive, real number that represents the desired minimal distance between the descriptors of different material classes. Thus, this method learns to map a microstructure image to a vector representation of a desired dimensionality, so that similar types of microstructures have descriptors that lay close together and different types of microstructures have descriptors that are separated from each other. More explanation on how a triplet network works, can be found in Supplementary S3. Further details on the training procedure can be found in Supplementary S4. All the results we report are obtained using the deep learning library PyTorch[16].

Throughout this work, we will only use two-dimensional descriptors, as using such low-dimensional descriptors allows for training more robust classification models afterwards[17]. The output of the triplet network is shown in Fig. 2, where each coloured dot represents the descriptor of a crop in the training set. Dots belonging to the same class have the same colour. Clearly, the neural network has been well optimized, as descriptors of the same material class lay close together and far away from the other classes.

In the second step, we need to assign each of the descriptors to the right material class. A key challenge in the quizzes is the possibility for the images to belong to none of the pre-defined classes, so that we need to recognize when a given microstructure image differs from the images in the dataset on which the model was

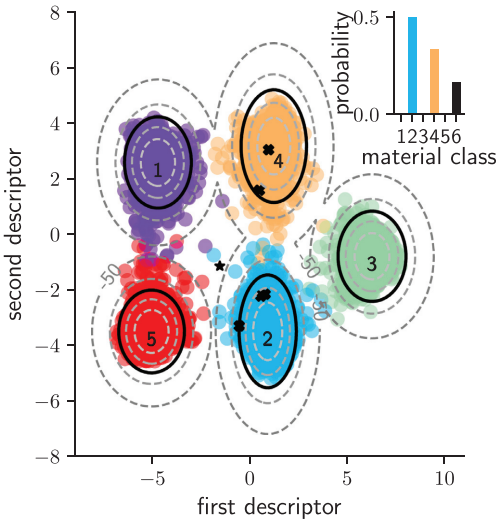


Fig. 2. Illustration of the used methodology. The coloured dots represent the descriptors of the crops used for training the model. Dots with the same colour belong to the same material class. The contours show the log-likelihood of the Gaussian mixture model that is fitted to these points. The black contour lines show the threshold for determining outliers. Points outside these contour lines are considered to belong to none of the pre-defined material classes. The five black crosses are crops from the same quiz image that are assigned to one of the classes, while the black star is a crop from that image that is considered an outlier. The resulting probability distribution for these crops is shown in the inset, where the "None of these" option is the sixth class. As most of the crops are assigned to class two, the model would predict the quiz image to belong to that class with a probability of 50%. Best viewed in colour.

trained. To do so, we fit a multivariate Gaussian distribution to each of the class centres of the descriptors using the Gaussian mixture model implementation in scikit-learn[18]. The resulting contours of the likelihood of the Gaussian mixture model are shown in Fig. 2. Crops that have a descriptor with a likelihood under a certain threshold are considered to belong to the "None of these" class. Details on the fitting procedure can be found in Supplementary S4 and details on the determination of the threshold can be found in Supplementary S5.

To be consistent with the training procedure, the model can only deal with images of size 200 × 200. During the evaluation of the quiz images, we therefore randomly select a number of crops from the image. After applying image augmentation to these crops, the microstructural descriptors are computed using the triplet network. These descriptors are then passed to the Gaussian mixture model to obtain for each crop a probability for each material class. This is shown in Fig. 2, where the black crosses and star represent crops belonging to the same image. After averaging the probabilities of the crops belonging to the same image, we obtain a probability distribution for the entire image. As averaging over more crops is preferable to obtain a meaningful probability distribution, we evaluate a thousand crops per image. By using a probability distribution rather than a hard decision, we can assess how certain the model is in its predictions.

To evaluate the performance of the proposed method, we hold two quizzes for both human participants and the deep learning model. For the first quiz, the human panel consists of 21 experienced metallurgists, to whom we will refer as "experts", and at 56 additional participants who don't necessarily have any experience in analysing microstructure images. We refer to this second group of participants as "non-experts". In the second quiz, 20 experts and 27 non-experts participated. The quiz was sent through a link and there was no time limit to answer the questions. Although we use the word "expert" for the metallurgists, we emphasize that especially for the second quiz, not everyone is equally familiar with the microstructures under consideration. Both panels were shown some representative example images for each material class, but they did not see the entire training set.

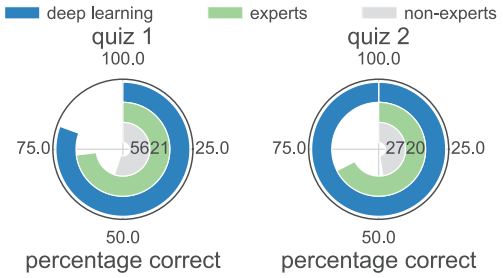


Fig. 3. Comparison of the classification accuracies of the panel of experts and the deep learning models for both quizzes. The numbers in the bars indicate the number of participants.

In Fig. 3, we compare the classification accuracy of the non-expert panel, the expert panel and the deep learning model. Unsurprisingly, the experts beat the non-experts by a large margin in both quizzes. The deep learning model achieved the highest accuracy in both cases. For the first quiz, the experts classified on average 73% of the images correctly, while the deep learning model achieves an accuracy of 81%. The best performing expert managed to obtain a score of 86%, outperforming the model by 5 percentage points. Still, we can conclude that the model is competitive with the best experts. For the second quiz, the experts obtained an average score of 67%, while the deep learning model classified all images correctly. The best expert in the panel also managed to obtain a perfect classification score. It is remarkable how well the model performs considering the small amount of data. Because deep learning models consider every single pixel in the image, they are able to discern very small details, which are important for this kind of complex microstructures. Code to reproduce the results can be found on microstructuredb.com/papers.

In Supplementary S7 and S8, we have included an overview for every quiz image of the probabilities assigned to each class by the machine learning algorithm. It is interesting to compare this to how the experts voted. It is clear for quiz 1 that both the human expert and the deep learning algorithm struggle with the "None of these" option. This option confuses the majority of experts in three out of the in total four wrongly classified images, whereas the deep learning algorithm fails because of this option in six out of the in total seven wrongly classified images. One such an image is shown in Fig. 4 (a), where we see a pearlitic-ferritic structure at a low magnification. We have also included a saliency map that shows the pixel-level predictions of the model. These maps are obtained by averaging the predictions of many crops using the procedure outlined in [11]. Since the training set only contains images that have a magnification of at least a factor $x2.5$ higher, the model has to extrapolate and fails to correctly recognize the microstructure. This is also reflected in the completely dark saliency map as the model fails to recognise any microstructural feature in the image. For the human experts, who can fall back on their much larger mental library of images, it is easier to recognize that this is indeed a low-magnification image of a pearlitic-ferritic material.

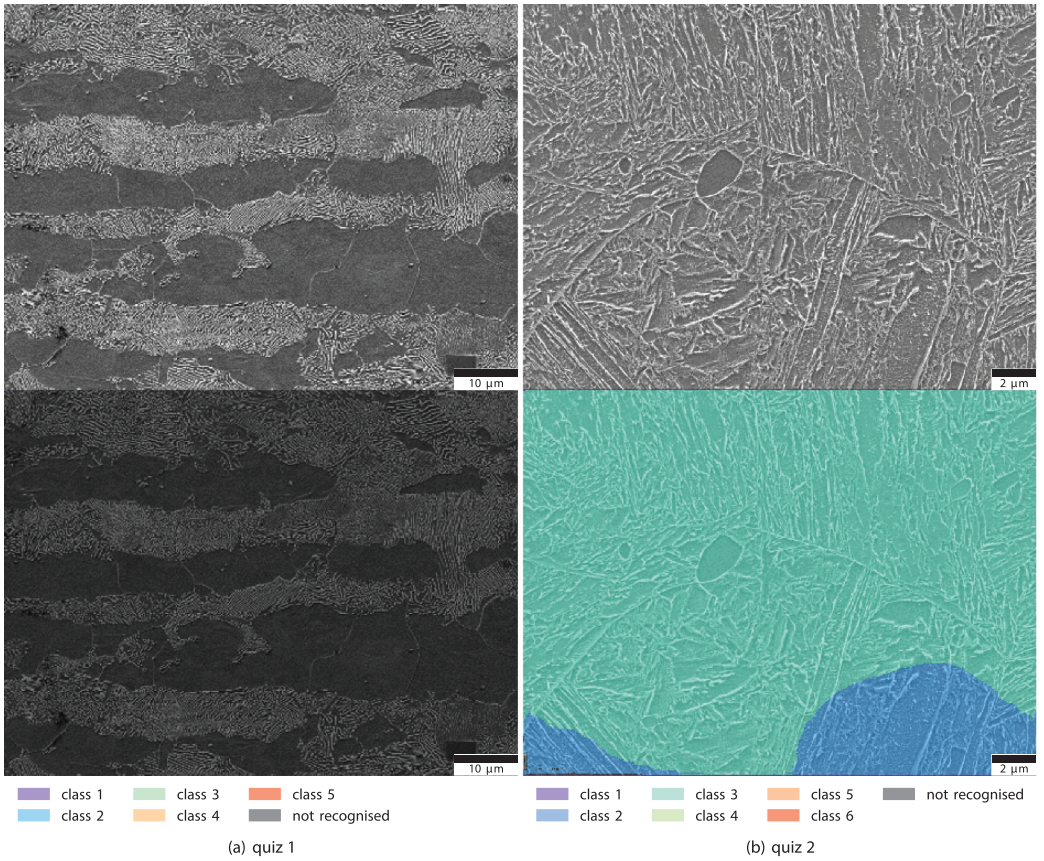


Fig. 4. Visual analysis of the model predictions using saliency maps. In both quizzes, there is a limited number of images where human experts outperform the deep learning algorithm, in contrast to the overall result. For each of the two quizzes, one of these images is shown here together with the saliency maps that provide pixel-level assignments of the model. The image on the left contains both ferrite and pearlite and hence belongs to class 4, whereas the model does not assign this image to one of the pre-defined classes. The saliency map is entirely dark, implying that no microstructural features are recognised. The image on the right features some sharp needle-like structures and hence belongs to class 3. The model correctly predicts this, although it is not very certain about its prediction, due to the presence of precipitates in the lower corners, which are typical features of materials from class 2. These regions are marked blue in the saliency map and hence assigned to class 2.

For the second quiz, we see that the machine learning model not only obtains a perfect classification score, but that is it also very confident in its predictions. The predictions in which it is least confident, are those for the images that belong to classes 3 and 5, which are the classes with the fewest training images. There is only one image where the human experts are clearly more confident in their prediction and this image shown in Fig. 4 (b). Due to the presence of some martensitic regions with a small amount of carbides in the lower right part of the image, it is indeed understandable for the model to assign some probability to class 2, which is the only class with such fine carbides. This is also shown in the saliency map, where the precipitate-containing regions are marked blue. In Supplementary S9 and S10, we discuss some more images and their saliency maps. In line with the findings in [11], we find that the model autonomously learns the importance of microstructural features such as the grain size, the grain shape and the presence of precipitates.

The results above are promising, as they show that even when small datasets of images are available deep learning models are capable of analysing microstructure images at least on par with human experts. As the deep learning methods will become even more performant, we anticipate that human experts will soon struggle to keep up with deep learning models and that such models will play a crucial role in the analysis of microstructure images.

In this report we illustrated how a deep learning model can be used to recognize structures even when only small datasets are available. We tailored practices from other fields in computer vision to the problem of microstructure recognition by introducing appropriate data augmentations and by explicitly including information about the magnification in the model. We showed how the models we trained can outperform a panel of experts and concluded that the use of deep learning is especially effective in the study of complex martensitic microstructures.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

M.L. and S.C. acknowledge financial support from OCAS NV by an OCAS-sponsored PhD position and by an OCAS-endowed chair at Ghent University, respectively. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.scriptamat.2020.10.026

References

[1] D.T. Fullwood, S.R. Niezgoda, S.R. Kalidindi, Acta Mater. 56 (2008) 942–948.
[2] A. Chowdhury, E. Kautz, B. Yener, D. Lewis, Computational Mater. Sci. 123 (2016) 176–187.
[3] B.L. DeCost, E.A. Holm, Computational Mater. Sci. 110 (2015) 126–133.
[4] J. Gola, J. Webel, D. Britz, A. Guitar, T. Staudt, M. Winter, F. Mücklich, Computational Mater. Sci. 160 (2019) 186–196.
[5] J. Schmidhuber, Neural networks 61 (2015) 85–117.
[6] D. Cireşan, U. Meier, J. Masci, J. Schmidhuber, Neural Networks 32 (2012) 333–338.
[7] S.M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, Sci. Rep. 8 (2018) 1–14.
[8] B.L. DeCost, B. Lei, T. Francis, E.A. Holm, Microsc. Microanal. 25 (2019) 21–29.
[9] F. Ajioka, Z.-L. Wang, T. Ogawa, Y. Adachi, ISIJ Int. 60 (2020) 954–959.
[10] B.L. DeCost, T. Francis, E.A. Holm, Acta Mater. 133 (2017) 30–40.
[11] M. Larmuseau, M. Sluydts, K. Theuwissen, L. Duprez, T. Dhaene, S. Cottenier, NPJ Comput. Mater. 6 (2020) 1–11.
[12] H. Tang, X. Chen, Y. Liu, Z. Lu, J. You, M. Yang, S. Yao, G. Zhao, Y. Xu, T. Chen, et al., Nature Machine Intelligence 1 (2019) 1–12.
[13] J. De Fauw, J.R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'xDonoghue, D. Visentin, et al., Nat. Med. 24 (2018) 1342–1350.
[14] E. Hoffer, N. Ailon, in: International Workshop on Similarity-Based Pattern Recognit., Springer International Publishing, Cham, 2015, pp. 84–92.
[15] F. Schroff, D. Kalenichenko, J. Philbin, in: IEEE Conference on Computer Vision and Pattern Recognit., IEEE, New York, 2015, pp. 815–823.
[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., in: Advances in neural information processing systems, Curran Associates, Brooklyn, 2019, pp. 8026–8037.
[17] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Springer, New York, 2001.
[18] S. Van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, PeerJ 2 (2014) e453.

B

List of Publications

Publications in international peer–reviewed journals

1. **Compact representations of microstructure images using triplet networks**
   Michiel Larmuseau, Michael Sluydts, Koenraad Theuwissen, Lode Duprez, Tom Dhaene and Stefaan Cottenier *NPJ Comput. Mater.*, **2020**, *16*, 1–11
   IF: 8.670. Number of citations: 1

2. **Race against the Machine: can deep learning recognize microstructures as well as the trained human eye?**
   Michiel Larmuseau, Michael Sluydts, Koenraad Theuwissen, Lode Duprez, Tom Dhaene and Stefaan Cottenier *Scr. Mater.*, **2020**, *193*, 33–37
   IF: 5.079. Number of citations: 0

Conference contributions

Poster presentations

1. **Deep learning representations of microstructures**
   Michiel Larmuseau, Maarten Cools-Ceuppens, Michael Sluydts, Toon Verstraelen and Stefaan Cottenier
   ML4MS 2018, Helsinki, Finland, May 3 – May 4, 2018

2. **Towards a more compact representation of microstructures using deep learning**
   Michiel Larmuseau, Maarten Cools-Ceuppens, Michael Sluydts, Toon Verstraelen and Stefaan Cottenier
   ICMPE 2018, Paris, France, September 25, 2018

3. **Towards active learning using reliable uncertainty estimates**
   Michiel Larmuseau, Maarten Cools-Ceuppens, Michael Sluydts, Toon Verstraelen and Stefaan Cottenier
   ML4MS 2019, Helsinki, Finland, May 6 – May 10, 2019

Master's thesis

Dissipation dynamics within quantum integrable models
Michiel Larmuseau
Master's thesis performed at the Center for Molecular Modeling (CMM), Ghent University, 2015–2016
Supervisor: prof. dr. Dimitri Van Neck

Bibliography

[1] G. Pivnyak, V. Bondarenko, and I. Kovalevska, *New Developments in Mining Engineering 2015: Theoretical and Practical Solutions of Mineral Resources Mining*. CRC Press, Boca Raton, 2015.

[2] T. F. Potts, "Patterns of trade in third-millennium bc mesopotamia and iran," *World Archaeology*, vol. 24, pp. 379–402, 1993.

[3] M. E. Weeks and H. M. Leichester, "Elements known to the ancients," *Journal of Chemical Education, Easton, Pennsylvania*, pp. 29–40, 1968.

[4] World Steel, "The uses of steel," 2020.

[5] R. Anstis, *Man of Iron-Man of Steel: The Lives of David and Robert Mushet*. Albion House, 1997.

[6] D. Jones and M. Ashby, *Engineering Materials 2: An Introduction to Microstructures and Processing*. Elsevier Science, 2012.

[7] B. Bregar, "Price keeping carbon fiber from mass adoption," 2014.

[8] G. B. Olson, "Computational design of hierarchically structured materials," *Science*, vol. 277, no. 5330, pp. 1237–1242, 1997.

[9] Wikipedia, "Machine learning," 2020.

[10] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science," *Apl Materials*, vol. 4, no. 5, p. 053208, 2016.

[11] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, pp. 1347–1358, 2019.

[12] S. Athey, "The impact of machine learning on economics," in *The economics of artificial intelligence: An agenda*, pp. 507–547, University of Chicago Press, 2018.

[13] S. Webb, "Deep learning for biology," *Nature*, vol. 554, 2018.

[14] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, p. 045002, 2019.

[15] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, pp. 354–359, 2017.

[16] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[17] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[18] T. Chung, *General continuum mechanics*. Cambridge University Press, 2007.

[19] W. Commons, "Stress-strain curve typical of a low carbon steel.," 2020. File: Stress_strain_ductile.svg.

[20] W. D. Callister, *Materials science and engineering an introduction*. John Wiley, 2007.

[21] R. G. Budynas and J. K. Nisbett, *Shigley's mechanical engineering design*. McGraw-Hill New York, 8 ed., 2006.

[22] G. Charpy, "Note sur l'essai des métaux à la flexion par choc de barreaux entaillés," *Mémoires et comptes rendus de la société des ingénieurs civils de France*, pp. 848–877, 1901.

[23] E. Izod, "Testing brittleness of steel," *Engineering*, vol. 76, pp. 431–2, 1903.

[24] D. Tabor, *The hardness of metals*. Oxford university press, 2000.

[25] R. L. Smith and G. Sandly, "An accurate method of determining the hardness of metals, with particular reference to those of a high degree of hardness," *Proceedings of the Institution of Mechanical Engineers*, vol. 102, pp. 623–641, 1922.

[26] J. Brinell, "Brinell's method of determining hardness and their properties of iron and steel." ii. cong. int. methodes d'essai, paris,"(translated to english by a. wahlberg)," *J. Iron Steel Inst*, vol. 59, pp. 243–298, 1901.

[27] Wikimedia Commons, "Vickers hardness test.," 2020. File: Vickers-path-2.svg.

[28] Wikimedia Commons, "Brinell hardness test.," 2020. File: BrinellHardness.svg.

[29] H. POLLOK, "Leeb-härteprüfung als alternative zu traditionellen verfahren: Umwertung der skalen," *QZ. Qualität und Zuverlässigkeit*, vol. 53, no. 4, pp. 76–79, 2008.

[30] E. Pavlina and C. Van Tyne, "Correlation of yield strength and tensile strength with hardness for steels," *Journal of materials engineering and performance*, vol. 17, pp. 888–893, 2008.

[31] W. F. McDonough and S.-S. Sun, "The composition of the earth," *Chemical geology*, vol. 120, no. 3-4, pp. 223–253, 1995.

[32] H. Bhadeshia and R. Honeycombe, *Steels: microstructure and properties*. Butterworth-Heinemann, 2017.

[33] A. Handbook, "Vol. 3: alloy phase diagrams," *ASM International, Materials Park, OH, USA*, vol. 2, p. 48, 1992.

[34] Y. Li, D. Raabe, M. Herbig, P.-P. Choi, S. Goto, A. Kostka, H. Yarita, C. Borchers, and R. Kirchheim, "Segregation stabilizes nanocrystalline bulk steel with near theoretical strength," *Physical review letters*, vol. 113, no. 10, p. 106104, 2014.

[35] Wikimedia Commons, "The iron-carbon phase diagram.," 2021. File: Diagramme_fer_carbone.svg.

[36] T. Baumeister and A. M. Sadegh, *Marks' standard handbook for mechanical engineers*, vol. 1. McGraw-Hill New York, 1978.

[37] Wikimedia Commons, "The cct curve of steel.," 2020. File: CCT_curve_steel.svg.

[38] A. Kumar, S. B. Singh, and K. Ray, "Influence of bainite/martensite-content on the tensile properties of low carbon dual-phase steels," *Materials Science and Engineering: A*, no. 1-2, pp. 270–282, 2008.

[39] K. to Metals AG, "Constant temperature transformation ttt curves," 2001. File: art14-p2.gif.

[40] R. Rottenfusse, E. E. Wilson, and M. W. Davidson, "Reflected light microscopy," 2020. File: reflectedfigure6.jpg.

[41] G. F. Vander Voort, *Metallography, principles and practice*. ASM international, 1999.

[42] Perdue university, "Scanning electron microscope," 2020. File: old%20rs%20graphics/sem2.gif.

[43] J. I. Goldstein, D. E. Newbury, J. R. Michael, N. W. Ritchie, J. H. J. Scott, and D. C. Joy, *Scanning electron microscopy and X-ray microanalysis*. Springer, 2017.

[44] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

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

[46] J. Webel, J. Gola, D. Britz, and F. Mücklich, "A new analysis approach based on haralick texture features for the characterization of microstructure on the example of low-alloy steels," *Mater. Charact.*, vol. 144, pp. 584 – 596, 2018.

[47] B. L. DeCost and E. A. Holm, "A computer vision approach for automated analysis and classification of microstructural image data," *Computational Mater. Sci.*, vol. 110, pp. 126 – 133, 2015.

[48] B. L. Adams, X. C. Gao, and S. R. Kalidindi, "Finite approximations to the second-order properties closure in single phase polycrystals," *Acta Materialia*, vol. 53, pp. 3563 – 3577, 2005.

[49] D. T. Fullwood, S. R. Niezgoda, and S. R. Kalidindi, "Microstructure reconstructions from 2-point statistics using phase-recovery algorithms," *Acta Materialia*, vol. 56, pp. 942 – 948, 2008.

[50] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[51] S. R. Niezgoda, Y. C. Yabansu, and S. R. Kalidindi, "Understanding and visualizing microstructure and microstructure variance as a stochastic process," *Acta Materialia*, vol. 59, pp. 6387 – 6400, 2011.

[52] J. Gola, D. Britz, T. Staudt, M. Winter, A. S. Schneider, M. Ludovici, and F. Mücklich, "Advanced microstructure classification by data mining methods," *Computational Materials Science*, vol. 148, pp. 324–335, 2018.

[53] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *The International Joint Conference on Neural Networks*, pp. 1918–1921, 2011.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[55] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[56] Peltarion, "2d-convolution-block," 2020. File: 2d_convolution_pa3.png.

[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[58] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[59] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[60] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, pp. 533–536, 1986.

[63] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, p. III–1139–III–1147, JMLR.org, 2013.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[66] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[67] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[71] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[72] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[73] F. Ajioka, Z.-L. Wang, T. Ogawa, and Y. Adachi, "Development of high accuracy segmentation model for microstructure of steel by deep learning," *ISIJ International*, pp. ISIJINT–2019, 2020.

[74] S. Torquato, "Statistical description of microstructures," *Annual review of Mater. Res.*, vol. 32, pp. 77–111, 2002.

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

[76] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognit.*, Springer International Publishing, Cham, 2015. 84–92.

[77] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, New York, 2015. 815–823.

[78] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer, New York, 2001.

[79] M. Larmuseau, M. Sluydts, K. Theuwissen, L. Duprez, T. Dhaene, and S. Cottenier, "Race against the machine: can deep learning recognize microstructures as well as the trained human eye?," *Scripta Materialia*, vol. 193, pp. 33–37, 2021.

[80] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[81] M. Larmuseau, M. Sluydts, K. Theuwissen, L. Duprez, T. Dhaene, and S. Cottenier, "Compact representations of microstructure images using triplet networks," *npj Computational Materials*, vol. 6, pp. 1–11, 2020.

[82] S. M. Azimi, D. Britz, M. Engstler, M. Fritz, and F. Mücklich, "Advanced steel microstructural classification by deep learning methods," *Scientific reports*, vol. 8, pp. 1–14, 2018.

[83] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[84] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *arXiv preprint arXiv:2004.11362*, 2020.

[85] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nat. Med.*, vol. 24, pp. 1342–1350, 2018.

[86] H. Tang, X. Chen, Y. Liu, Z. Lu, J. You, M. Yang, S. Yao, G. Zhao, Y. Xu, T. Chen, *et al.*, "Clinically applicable deep learning framework for organs at risk delineation in ct images," *Nature Machine Intelligence*, vol. 1, pp. 1–12, 2019.

[87] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[88] E. W. Grafarend, *Linear and nonlinear models: fixed effects, random effects, and mixed models*. de Gruyter, Berlin, 2006.

[89] D. L. McDowell and R. A. LeSar, "The need for microstructure informatics in process-structure-property relations," *MRS Bulletin*, vol. 41, no. 8, pp. 587–593, 2016.

[90] B. Yucel, S. Yucel, A. Ray, L. Duprez, and S. R. Kalidindi, "Mining the correlations between optical micrographs and mechanical properties of cold-rolled hsla steels using machine learning approaches," *Integrating Materials and Manufacturing Innovation*, pp. 1–17, 2020.

[91] X. Li, Y. Zhang, H. Zhao, C. Burkhart, L. C. Brinson, and W. Chen, "A transfer learning approach for microstructure reconstruction and structure-property predictions," *Scientific reports*, vol. 8, pp. 1–13, 2018.

[92] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, and L. Song, "Material structure-property linkages using three-dimensional convolutional neural networks," *Acta Materialia*, 2018.

[93] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

[94] B. Efron, *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.

[95] A. International, *Standard test method for Brinell hardness of metallic materials*. ASTM International, 2012.

[96] E. C. Bain and H. W. Paxton, "Alloying elements in steel," *1966, 291 P. AMERICAN SOCIETY FOR METALS, METALS PARK, OHIO*, 1966.

[97] R. Grange, C. Hribal, and L. Porter, "Hardness of tempered martensite in carbon and low-alloy steels," *Metallurgical Transactions A*, vol. 8, no. 11, pp. 1775–1785, 1977.

[98] K. de Haan, Z. S. Ballard, Y. Rivenson, Y. Wu, and A. Ozcan, "Resolution enhancement in scanning electron microscopy using deep learning," *Scientific Reports*, vol. 9, pp. 1–7, 2019.

[99] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian super resolution," *IEEE Transactions on Image Processing*, vol. 20, pp. 984–999, 2010.

[100] M. Lee, J. Cantone, J. Xu, L. Sun, and R.-h. Kim, "Improving sem image quality using pixel super resolution technique," in *Metrology, Inspection, and Process Control for Microlithography XXVIII*, vol. 9050, p. 90500U, International Society for Optics and Photonics, 2014.

[101] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th international conference on computer vision*, pp. 349–356, IEEE, 2009.

[102] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, vol. 22, pp. 56–65, 2002.

[103] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[104] R. Brunelli, *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[105] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1858–1865, 2008.

[106] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, Curran Associates, Inc., 2014.

[107] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, pp. 2234–2242, Curran Associates, Inc., 2016.

[108] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[109] W. Burger and M. J. Burge, *Principles of digital image processing: core algorithms*. Springer Science & Business Media, 2010.

[110] B. L. DeCost, M. D. Hecht, T. Francis, B. A. Webler, Y. N. Picard, and E. A. Holm, "Uhcsdb: Ultrahigh carbon steel micrograph database," *Integrating Materials and Manufacturing Innovation*, vol. 6, pp. 197–205, 2017.

GHENT
UNIVERSITY

This research was enabled by an OCAS-endowed PhD position.

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO, and the Flemish Government – department EWI.