ISBN 978-82-326-5694-3 (printed ver.)
ISBN 978-82-326-5329-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

NTNU
Norwegian University of
Science and Technology

Kurdistan Chawshin

Doctoral thesis

Doctoral theses at NTNU, 2021:316

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
Philosophiae Doctor
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU

Doctoral theses at NTNU, 2021:316

Kurdistan Chawshin

Automated Lithology Classification Employing Whole Core CT-Scans

NTNU
Norwegian University of
Science and Technology

Kurdistan Chawshin

Automated Lithology Classification Employing Whole Core CT-Scans

Thesis for the degree of Philosophiae Doctor

Trondheim, October 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU

Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering
Department of Geoscience and Petroleum

ISBN 978-82-326-5694-3 (printed ver.)
ISBN 978-82-326-5329-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (electronic ver.)

Doctoral theses at NTNU, 2021:316

PRINTED IN
NORWAY
NO - 1598

NORDIC SWAN ECOLABEL
Printed matter
2041 0731

Printed by Skipnes Kommunikasjon AS

Preface

The work presented in this doctoral thesis has been carried out in the period of September 2018 – September 2021 at the Department of Geoscience and Petroleum at Norwegian University of Science and Technology (NTNU) in Trondheim. The research is accomplished under supervision of associate professor Carl Fredrik Berg from the Department of Geoscience and Petroleum and co-supervision of professor Damiano Varagnolo from the Department of Engineering Cybernetics at NTNU. The project is part of the BRU21 (Better Resource Utilization in the 21st century) project at NTNU. BRU21 is NTNU's research and innovation program in digital and automation solutions for the oil and gas industry. The data and financial support for this project, is provided by Equinor Norway, and Dr. Olivier Lopez has been our main contact person at Equinor.

The aim of this study is to utilize the information content of whole core CT-scans, in combination with well logs and core analysis data, to automate lithology classification tasks and to estimate transport properties through application of machine learning algorithms.

Within this study, three journal papers and one peer reviewed conference paper were prepared for publication to represent different parts of the conducted research. The first three papers introduce workflows for automatic classification of lithology employing 2D (**paper I** and **paper II**) and 3D (**paper III**) core CT images, while **paper IV** focuses on the automatic estimation of porosity using 2D CT-scan images.

Acknowledgements

This research is a part of BRU21–NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21) supported by Equinor in Norway. I would like to thank Equinor for providing the challenge, financial support, and data employed in this thesis.

Throughout these three years I have received a great deal of support and assistance.

First and foremost I would like to express my gratitude to my supervisors associate professor Carl Fredrik Berg and professor Damiano Varagnolo, from Norwegian University of Science and Technology (NTNU), for their invaluable guidance, continuous support, and patience during these three years. Thank you for your kindness, help, and the fruitful discussions during our regular meetings.

A special thanks goes to Dr. Olivier Lopez at Equinor for providing the data, support, and valuable input during our progress status meetings.

I would also like to thank PhD student Andres Gonzalez and Associate professor Zoya Heidari, from the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin, for valuable discussions and their contribution to the first paper.

Last but not least I like to thank my family for their encouragement and enduring support. A warm and special thanks to my husband for his continued love, friendship, valuable feedback, and understanding during these three years that made the completion of this thesis possible. Your love and unconditional support for me and our little daughter was always around and heartwarming especially when I had to stay in the office for long hours. Thanks for being such a special person in my life!

Abstract

This thesis evaluates the application of artificial intelligence to detect rock properties from whole core computed tomography (CT) scan images of wells on the Norwegian continental shelf. Whole core CT-scan images provide millimeter scale 3D information on the composition and internal structure of the imaged reservoir rocks.

Lithology classification is crucial to better understand the depositional environment and properties of the reservoir rocks. Reservoir transport properties such as porosity, permeability, and water saturation are closely correlated with lithology through lithological properties. Therefore, lithology classification is an essential step in formation evaluation and reservoir characterization processes. In this thesis, we have evaluated the possibility of automated lithology classification and transport property estimation using whole core CT-scan images, in combination with well logs and routine core analysis data. The automation task was accomplished by application of machine learning algorithms, i.e., support vector machines (SVM) and convolutional neural networks (CNN).

More precisely, three workflows are proposed for classification of lithology. In the first workflow (**paper I**), first-order statistical features and textural gray-level co-occurrence matrix (GLCM) features are extracted from pre-processed and transformed 2D cross-sectional CT-scan images. These features are then used as an input to train an SVM classifier to classify lithofacies based on ground truth classes that were derived from manual core descriptions. In a second approach in the same paper, a principal component analysis (PCA) step is added before training with two purposes: first, to eliminate collinearity among the features and second, to investigate the amount of information needed to differentiate the analyzed images. The second workflow for lithology classification (**paper II**) utilizes 2D image slices directly as input to a CNN classifier to learn the relationship between convolution-derived features and expert-derived (manual) lithofacies classes. Finally, the third

workflow (**paper III**) employs the 3D CT-scan images to train a 3D CNN classifier. To evaluate the generalization capabilities of the trained classifiers, they were employed to predict lithofacies classes on a set of unseen images.

The acquired lithology prediction results revealed that SVM and CNN classifiers trained on 2D images show lithology-dependent accuracy, and lithofacies with similarities in the texture and grayscale attenuation values are confused by these classifiers. More importantly, additional analyses of the porosity-permeability trends indicate that misclassified lithofacies share similar transport properties. These findings can be helpful in understanding the similarities between various lithofacies classes and their corresponding transport properties. In **paper II** the degree of confusion from prediction results and porosity-permeability trends were utilized as a tool to cluster similar lithofacies classes into coarser rock classes as a post-classification step to refine the acquired results. Overall, the obtained lithofacies classification results based on 2D CT images reveal higher performance of the CNN classifier in predicting unseen images compared to the SVM classifier. However, apart from the inherent differences in the statistical generalization capabilities of these algorithms in general, we should consider that these two approaches use different types of data to classify lithofacies. 2D cross-sectional images and the features manually extracted from them might not be completely representative of the 3D variations in core data.

The CNN classifier trained on 3D images (**paper III**) shows higher generalization capabilities compared to the CNN classifier trained on 2D image slices. This might indicate the added value of the full 3D information for CNN in extracting relevant features and correct identification of lithofacies classes. However, one should note that, due to computational limitations, only a small interval of the well consisting of three (out of twenty) lithofacies classes was used to train the 3D classifier. To ensure a fair comparison, the 3D classifier should have been trained and evaluated on all the lithofacies classes. This was not possible during this PhD study due to lack of enough computational resources.

The possibility of automatic porosity estimation from 2D image slices was also investigated in this thesis (**paper IV**), where an end-to-end CNN regression model was trained to learn from routine core analysis (RCA) porosity measurements. To characterize the capability of such approach, we compared the performance of this model during training with a linear regression model trained to learn the relationship between the average attenuation of the same 2D images and RCA porosity. The training results showed the superior performance of the CNN model, indicat-

ing the importance of accounting for the distribution of the gray-level attenuation present in the CT scan images for porosity estimations. The trained CNN model was then used to predict porosity on the unseen images and to populate two wells with millimeter scale porosity values. Comparison of the predicted porosity results against the RCA porosity measurements and total porosity log, calculated from the density log, shows that the predicted porosity values using the proposed CNN model are well correlated with the core plug measurements and porosity log. More importantly, the proposed approach can provide accurate millimeter scale porosity estimations, while the total porosity log is averaged over an interval and do not show such fine scale variations. Thus, the proposed method can be employed to calibrate the porosity logs at the relevant scale, thereby reducing the uncertainties associated with indirect calculations of the porosity from such logs.

In summary, this study confirms the value of whole core CT-scans, as digital representations of the reservoir rocks, in automated lithology classification and transport property estimations. However, there are limitations and uncertainties associated with imbalanced distribution of the lithofacies classes in the training dataset, image artefacts, image complexities and the quality of manual core descriptions that can negatively affect the training process and generalization capabilities of the proposed models.

Thesis structure

The presented doctoral work is organized in two parts: Part I provides a summary of the conducted work. This part consists of eight chapters. Chapter 1 describes the motivation and objectives of the thesis. The fundamentals of coring, the existing lithology classification methods, and fundamentals of CT-scanning are described in Chapter 2. The third chapter provides a detailed literature review on the application of artificial intelligence in lithology classification and transport property estimations. Chapter 4 presents the theory of employed methods, including wavelet transform, feature extraction, and machine learning algorithms. The material used in this study will be described in Chapter 5. This will include whole core CT scan data, the lithology of the studied wells, and porosity-permeability measurements from routine core analysis. The employed methodologies for depth shift, image pre-processing, lithology classification and transport property estimation are described in Chapter 6. The obtained results from individual papers and their relations are presented and discussed in Chapter 7. Finally, Chapter 8 provides the main concluding comments.

Part II presents the four scientific papers as the main contributions to the doctoral work.

List of publications

1. **K. Chawshin**, C.F. Berg, D. Varagnolo, A. Gonzalez and Z. Heidari, and O. Lopez, 2021. Classifying Lithofacies from Textural Features in Whole Core CT-Scan Images. SPE Reservoir Evaluation and Engineering. SPE-205354-PA. 24(02), 341-357.

2. **K. Chawshin**, C.F. Berg, D. Varagnolo, and O. Lopez, 2021. Lithology Classification of Whole Core CT Scans Using Convolutional Neural Networks. SN Applied Sciences, 3(6), 1-21.

3. **K. Chawshin**, C.F. Berg, D. Varagnolo, and O. Lopez, 2021. A Deep Learning Approach for Lithological Classification Using 3D Whole Core CT Scan Images. SPWLA 62^{nd} Annual Symposium Boston, May 17-20.

4. **K. Chawshin**, C.F. Berg, D. Varagnolo, and O. Lopez, 2021.Automated Porosity Estimation using CT-scans of Extracted Core Data. Submitted to the Computational Geosciences.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

2D	Two-dimensional
3D	Three-dimensional
AC	Compressional wave slowness
ACS	Shear wave slowness
AI	Artificial intelligence
ANN	Artificial neural network
API	Application programming interface
BPNN	Back-propagation neural network
CALI	Caliper log
CNN	Convolutional neural network
CT	Computed tomography
CWT	Continuous wavelet transformation
DEAX	Dual-energy X-ray absorptiometry
DECT	Dual-energy CT scanning
DEN	Density log
DICOM	Digital imaging and communications in medicine (a data format for CT images)

DRA	Digital rock analysis
ERM	Empirical risk minimization
EVD	Eigenvalue decomposition
FL	Fuzzy logic
Fm.	Formation
FMS	Formation micro resistivity
GA	Genetic Algorithm
GAP	Global average pooling
GLCM	Gray-level co-occurrence matrix
GPU	Graphics processing unit
GR	Gamma ray log
H(n)	High-pass filter
HL	Horizontal detail image
L(n)	Low-pass filter
L1	Wavelet decomposition Level 1
L2	Wavelet decomposition Level 2
L3	Wavelet decomposition Level 3
LDA	Linear discriminant analysis
LH	Vertical detail image
LL	Approximation image
LSSVM	Least square support vector machine
mD	Millidarcy
MLP	Multilayer perceptron

NMR	Nuclear magnetic resonance
NPH	Neutron porosity log
PCA	Principal component analysis
PEF	Photoelectric absorption log
PHI	Neutron and density porosity average
RBF	Radial basis function
RCA	Routine core analysis
RDEP	Deep resistivity
ReLU	Rectified linear unit
SCAL	Special core analysis
SEG	Society of exploration geophysicists
SEM	Scanning electron microscopy
SP	Spontaneous potential
SSGMM	Semisupervised Gaussian Mixture Model
STL	Statistical learning theory
VC	Vapnik-Chervonenk dimension

Symbols

α	Learning rate
$\bar{X}$	Mean matrix
$\bar{x}$	mean
$\bar{x}_{c_i}$	Coarsened average attenuation at depth i
η	A small number
γ	Gaussian kernel parameter

$\hat{y}$	Predicted value
λ	Eigenvalue
μ	Attenuation coefficient
μ_c	Center average attenuation in cross-sectional images
μ_G	GLCM mean
μ_i	Inner average attenuation in 2D image slices
μ_o	Outer average attenuation
Φ_t	Total porosity
ψ	Mother wavelet
ψ_H	Haar mother wavelet
ρ	Correlation
ρ_b	Bulk density
ρ_e	Electron density
ρ_{fl}	Fluid density
ρ_{int_i}	Interpolated density at depth i
ρ_{ma}	Matrix density
σ	Standard deviation
σ^2	Variance
σ_G^2	Variance of the intensity of all reference pixels in the relationships that contributed to the GLCM
τ	Constant
a	Scale factor
A_1	Approximation wavelet coefficient of first decomposition level
A_2	Approximation wavelet coefficient of second decomposition level

A_3	Approximation wavelet coefficient of third decomposition level
B	Mean-centered matrix of data points
b	Translation factor
$B^\top$	Transposed mean-centered matrix of data points
b_i	Bias vector of the i^{th} fully connected layer
C	Covariance matrix
c	SVM misclassification constant
C_b	Core barrel coupling cutoff
C_h	High density material cutoff
C_m	Missing cutoff
D	Contrast
d	Depth
D_1	Detail wavelet coefficient of first decomposition level
D_2	Detail wavelet coefficient of second decomposition level
D_3	Detail wavelet coefficient of third decomposition level
E	Energy
f	Bounded function, e.g., a signal
$f(x_i)$	Predicted value for i^{th} data point
F_b	Core barrel coupling flag
F_h	High density flag
F_{ms}	Missing flag in image slices
F_m	Missing flag in cross-sectional images
g	Non-linear activation function

$G(i, j)$	Co-occurrence matrix representing relative frequency of two pixels with intensities i and j
H	Hypothesis space
h	Length of the X-ray path through the object
I	attenuated X-ray intensity
i, j	Pixel intensities
I_0	initial X-ray intensity
K	Kurtosis
k	Klein-Nishina coefficient
L	Loss function
l	Total number of training data points
L_c	Cross entropy loss
L_{emp}	Empirical error
M	Total number of depth points
M_o	Mode
MAE	Mean absolute error
MSE	Mean squared error
N	Total number of pixels
o_i	Output vector of the i^{th} convolutional layer
P	Total number of grayscale values of the GLCM
$P(x, y)$	Probability distribution of training samples
Q	Cost function
q	VC (Vapnik-Chervonenk) dimension
R	A set of real numbers

R_{ta}	Apparent true resistivity
S	Skewness
V	Eigen vector
w	Weight
w^+	Updated learnable parameter
w_i	Kernel weight of the i^{th} convolutional layer
X	Matrix of data points
x	Gray-level attenuation values
x_i	The i^{th} data point
y	Actual values/classes
Z	Average atomic number
z_1	First principal component
z_2	Second principal component
Z_{eff}	Effective atomic number
z_i	Output vector of the i^{th} fully connected layer

Part I: Summary

Chapter 1. Introduction

1.1 Motivation

An essential step in building a reliable reservoir model is describing the lithology to better understand the depositional environment and the distribution of rock properties. Reservoir properties such as porosity, permeability and water saturation are closely correlated with lithology through lithological properties such as mineralogical composition, grain size, sorting, and cementation. Traditional lithology classifications relying on well log interpretations and core descriptions provide the base for reservoir parameter calculations, thereby helping the geologists and reservoir engineers to identify hydrocarbon and/or water bearing formations. Well logs provide continuous information by covering the entire geological formation of interest. However, such interpretations are highly uncertain due to ambiguities in measurements, mineralogical complexities, wellbore environment, and other factors that can affect the log responses (Salehi and Honarvar 2014; Xie et al. 2018). Moreover, heterogeneities occurring at smaller scales than the log resolution, such as thin layers, are neglected in well log interpretations. Such heterogeneities can be detected by visual inspection of drill cuttings and extracted cores. However, lithology inference from cuttings is always uncertain due to uncertainties related to the depth of the retrieved cuttings (Salehi and Honarvar 2014). Additionally, lithology descriptions by visual inspection of whole cores are significantly subjective and time consuming. On the other hand, substantial capital investments have already gone into extracting the core data. Therefore, automated lithology classification and associated core analysis is considered as a key technology enabling return on investments and to enhance the overall decision processes.

As mentioned, transport properties such as porosity and permeability are highly correlated with lithology variations within the reservoir. Porosity is commonly estimated by indirect calculations from well log responses such as density, neutron, and sonic logs at centimeter scale acquisition resolution. There are high uncertainties in these types of porosity estimations since the log responses are affected by other properties in addition to porosity, e.g., lithology, type of fluids in the pore space, wellbore environment and type of drilling mud. Therefore, log-derived porosity interpretations need to be corrected and calibrated against core plug porosity measurements. Moreover, these estimations neglect higher resolution (below log scale) porosity variations that are critical for proper description of reservoirs with thin layers and fine scale heterogeneities. Acceptable local porosity estimations can be provided by laboratory measurements of core plugs extracted from whole cores. However, core plugs, normally sampled once per foot, might not provide representative data in heterogeneous reservoirs with millimeter scale porosity variations. More frequent sampling is necessary in this kind of reservoirs, which is laborious and cost consuming. Continuous core (millimeter) scale porosity estimations can provide a valuable tool for calibration of indirect log-derived porosity calculations, and they can help in identifying the core plug locations most valuable for core analysis.

Whole core CT-scan images provide millimeter scale 3D information on the composition and internal structure of the imaged reservoir rocks. Significant portions of the extracted cores are being imaged as an integrated part of the core handling workflows (Mena et al. 2015). Traditionally, CT images have been employed to study cores with respect to rock characterization and to evaluate drilling locations for core plugs. This has included inspection of 2D image slices at fixed cross-sections and videos of the 3D data. The full 3D information has seldom been exploited. Recent improvements in CT scanning technology has enabled more quantitative use of CT generated image data (Lopez et al. 2016), which has lead to a revived interest in CT-imaging in reservoir characterization workflows.

Available information about reservoir rocks occurs at non-overlapping scales, e.g., core plugs at the centimeter scale, logs at sub-meter scale, to seismic data at the meter-scale. Integration and one-on-one calibration between such data sources is essential for proper evaluation of the reservoirs. Extracted whole cores enable connecting the core plug scale to the log scale, thereby enabling integration of transport properties measured on core samples to down-hole log measurements.

The valuable information provided by core CT-scan images together with the fact that these images are stored digitally make them a proper candidate to be incorporated in the reservoir characterization workflows, thereby enhancing their value in the operational settings and facilitating the automated core classification process. However, a main question that arises for the industry is that whether the information content of whole core CT scans is enough to automatically classify different lithofacies classes and to estimate transport properties.

1.2 Objectives and scope of the PhD study

This PhD thesis evaluates the possibility of automated lithology classification and transport property estimation using the information content of whole core CT-scan images. To evaluate this possibility, this project aimed to develop routines and workflows to exploit the possibilities in whole core CT-scan images in combination with well log and core analysis data. It addressed rock typing based on automated image analysis routines, and investigated the possibilities for machine learning procedures based on the CT images. In other words, this project had a twofold objective: 1) to investigate whether the information content of core CT-scan images is enough to classify lithology and estimate transport properties, and 2) to develop workflows for automated lithology classification and transport property estimations based on the information content of 2D and 3D core CT images in combination with well logs and core analysis data.

These workflows include employing supervised learning algorithms, where the network is either trained to classify the lithology from 2D and 3D CT images based on a given lithofacies description, or it is trained to predict transport properties given properties measured on core plugs. In this project, we specifically considered porosity estimation using the porosity derived from routine core analysis measurements as the training data. Note that the lithology classification and porosity estimation objectives are linked together as porosity is closely correlated with lithology variations within the reservoir.

Chapter 2. Background

This chapter presents the fundamentals of coring, traditional lithology classification methods, and fundamentals of the CT scanning processes.

2.1 Fundamentals of coring

A proper reservoir characterization describes the reservoir as accurately as possible employing a variety of methods at different scales, from seismic, well testing and logging scales to core scale, cuttings, thin sections, and photomicrographs of minerals. The recovered cores from drilled wells are the only way to directly characterize rock properties in the laboratory, through analyzing the extracted core plugs. Coring and core analysis as an integral part of formation evaluation provide valuable information on lithology, grain density, transport properties, fluid saturation, fluid distribution, and multiphase fluid flow properties at a scale which is unavailable from log measurements and productivity tests (Al-Saddique et al. 2000). Moreover, core analysis measurements such as porosity can be utilized to calibrate log responses, e.g., density, neutron, and sonic logs, used to calculate porosity.

Representative and reliable core analysis measurements require high quality and undamaged core data. Fundamental coring methods consist of three categories including conventional coring, wireline continuous coring, and sidewall coring.

Full-diameter cores, also known as whole cores, provide continuous sections of rocks extracted from subsurface formations using conventional coring. This method is applied at the time of drilling and utilizes a rotary coring bit attached to the drill-string. The rotary coring bit uses diamonds or tungsten for cutting, and unlike the drillbit it has a hollow center that captures a cylinder drilled out of the penetrated rock. The drilled core can be then transferred to the surface as a single cylindrical piece of rock. Core bits are selected based on the formation types and hardness of the rocks to be cored. The core barrel consists of an inner and outer barrel and a core catcher. The inner and outer barrels are attached to a swivel that enables the inner barrel to remain stationary, while the outer barrel is rotated by the drillstring as the core is cut. The core catcher, located inside the core barrels, keeps the core from slipping out of the core barrel when the core is retrieved to the surface (McPhee et al. 2015).

The length of conventional core barrels can vary from around 30 to over 400 feet (9.14 to 122 m) (McPhee et al. 2015). The core diameter depends on the hole size, and it can vary from 1.75 to 5.25 in. (4.45 to 13.34 cm). Generally, the smaller diameter whole cores are likely to be exposed to more drilling mud invasion and more potential damage compared to the cores with larger diameter. Mud invasion, especially oil-based mud filtrates, can result in significant alteration of native wettability and formation fluid saturations in the drilled cores. To minimize filtrate invasion, low-invasion core bits are often employed. These are designed for fast cutting, thereby reducing the mud exposure time and subsequent mud invasion.

Once the coring process in completed, the drillstring is pulled up to the surface and the coring bit, the barrel and the catcher are removed. Then, after attaching the drillbit to the drillstring, the drilling can start again. Conventional coring operations often provide the best rock samples for laboratory measurements. However, this process is quite expensive due to suspension of drilling, time required to cut and recover whole cores, and the cost of the coring and core handling equipment. Therefore, this type of coring is usually only performed at the reservoir interval.

Wireline continuous coring is also performed at the time of drilling. However, it does not require a conventional trip for each retrieval of core or pipes to the surface. In this method, the coring and drilling modes can be easily switched to each other using slick line/wireline. In coring mode, the inner tube assembly is forced down into the outer tube assembly by slick line, and mud circulation starts between the core barrel assembly and the drill collar to secure the inner tube assembly to the outer tube assembly so that the core cutting can start. Once the core-drilling operation is accomplished, the acquired rock sample is brought up to the surface through the outer tube assembly using the slick line. Then, switching to the drilling mode, the inner drilling assembly is run into the hole again to proceed with drilling (Ashena and Thonhauser 2018).

Note that the main difference between conventional coring and wireline continuous coring is that in continuous coring the core bit, outer tube assembly and drill pipes are not pulled up to the surface, and they are kept in the hole for both coring and drilling modes. The main advantage of this method is that the time and cost of the coring operation is reduced since the core assemblies are not removed by conventional tripping. However, since the cores are tripped to the surface through the drill pipe, size of the retrieved cores is rather small, i.e., ranging from 1.73 to 3 in. (4.4 to 7.62 cm). Therefore, the conventional method is preferred for long coring of a single formation (Ashena and Thonhauser 2018).

Sidewall cores are acquired perpendicular to the wellbore, and by tools attached to wireline. This type of cores are acquired to minimize coring costs or to obtain reservoir rock samples on formation intervals with production potential. These intervals are either not cored conventionally or lost during core recovery. Sidewall coring is performed after drilling, and the potential intervals are identified by interpretation of well log responses.

Sidewall coring can be applied using percussion and rotary methods. In percussion coring a series of hollow bullets are fired into the formation using an explosive charge. The bullets, along with the core plugs, are then brought back into the coring tool and the tool is pooled up to the surface. Typical percussion sidewall plug sizes are ranging from 0.75 to 1 in. (1.9 to 2.5 cm) diameter by 1 in. (2.5 cm) long. Employing this method, up to 66 plugs can be recovered on a single run (McPhee et al. 2015).

A percussion coring system is quick and cost-effective. However, the bullets can break the cores in hard formations, resulting in empty retrieved core barrels. Sampling in soft formations is easier, although the formation might get consolidated, and the barrels may become so deeply embedded that they can not be extracted (Agarwal et al. 2013). Due to the impacts of bullets on the percussion sidewall cores, these cores are not representative for porosity and permeability measurements. The main application of such samples is therefore limited to lithological description, grain size, palynology, and palaeontology.

Rotary sidewall tools use a series of rotary coring bits to drill a core plug from the borehole wall. The core is then detached and pulled into the core holding area inside the tool body. This process can be repeated until the core catching apparatus is full. Since the mechanical distortion of the rock samples is eliminated in this method, reliable porosity and permeability data can be obtained from rotary sidewall cores. The rotary sidewall cores are typically 0.92 in. (2.33 cm) diameter by 1 (2.5 cm) to 1.5 in. (3.81 cm) long (McPhee et al. 2015), which are smaller than the typical laboratory core plugs cut from conventional cores. Insufficient volume of rotary cores, smaller than the standard core holders for routine and special core analysis (SCAL), may result in substandard core analysis results. Recent developments in sidewall coring systems can take much larger plugs that are comparable with typical SCAL plugs (i.e., 1.5 in. (3.18 cm) diameter by up to 3 in. (7.62 cm) long).

2.2 Lithology and lithology identification

Lithology is a description of the physical characteristics of a rock, including composition, color, grain size, and texture. As lithology controls the distribution of reservoir properties (e.g., porosity, permeability, and water saturation), a good understanding of its type and spatial variation is of great importance for an effective reservoir characterization. Direct understanding of the subsurface lithology can be obtained through observation of the extracted whole cores or sidewall cores, but these are costly and rarely cover the entire stratigraphic interval of interest (Chang et al. 2000). Moreover, manual core description is a time-consuming process. The lithology in non-cored well intervals can be identified through examination of drill cuttings that are brought back to the surface by the circulating drilling mud. Drill cuttings are the basis and the only opportunity for direct identification of subsurface lithology in non-cored well intervals. The cuttings are collected at predetermined depth or time intervals, calculated to correspond to regular changes in formation depth (e.g., every 5 to 10 m). The retrieved cuttings are either retained as "wet samples" or washed, dried and preserved as "dry samples". Wet samples are described and examined under microscope to provide information on grain properties, diagenesis, porosity type, permeability (qualitatively described as tight, slightly permeable, highly permeable), and presence of hydrocarbons (Darling 2005). The cuttings are a valuable source of information. However, the retrieval depth is always uncertain, and they are not suitable for accurate measurements of porosity and permeability due to their small size.

Various well log responses can be used to indirectly interpret subsurface lithology. An advantage of logs is that they usually cover the entire interval of interest, and they can be obtained where coring is impossible. Moreover, considering the total costs, logging is cheaper than coring. A combination of cored intervals and log data can provide the sedimentologist with good source of information for lithology classification (Serra and Abbott 1982).

The commonly available wireline logs, such as caliper (CALI), spontaneous potential (SP), gamma ray (GR), formation density (DEN), neutron (NPH), photoelectric absorption (PEF or P_e), and sonic or acoustic (AC) logs are considered as the most appropriate logs for lithology identification. Borehole imaging tools, especially Formation MicroScanner (FMS) borehole wall images provide invaluable detailed information on the bedding and sedimentary structures, but they are less commonly available (Ellis and Singer 2007).

The caliper log measures the diameter of the borehole. When a hole is the same diameter as the bit size, it is called "on gauge", which is typical of non-permeable and well consolidated formations. The caliper logs show diameters larger than the bit size in "washed out" intervals. These intervals are characteristics of certain lithology such as shales, unconsolidated sands, coals, and salt formations drilled with fresh water. A borehole diameter smaller than the bit size can be an indication of mud-cake build-up in permeable zones. Another frequent cause is related to formations swelling in presence of smectite in the clay mineral mixture, so called "tight spots". Smectite is a swelling clay that can take the water from the drilling mud, expand, slough, and collapse in to the hole (Rider 1986).

Spontaneous potential is a record of direct current voltage that spontaneously develops between a movable electrode in the well and a fixed electrode located at the surface (Ellis and Singer 2007). The SP response is affected by salinity differences between mud filtrate and formation water in permeable intervals. Presence of a conductive drilling mud is necessary to create SP response. Therefore, SP log cannot be employed in boreholes with non-conductive (oil based) drilling mud or in air-filled holes. The SP log interpretation relies on first recognizing a straight "shale baseline", where the log shows a relatively constant response. The SP readings are measured relative to this shale baseline. In sandstone reservoirs, the permeable zones are identified whenever there is a deflection from this line. The deflection is negative for a normal salinity contrast (i.e., the ionic concentration of the well bore fluid is less than the formation water). Tight rock types (i.e., tight sandstones and tight carbonates) result in poor or no response on the SP curve because of no ion exchange in the absence of permeable beds.

Gamma ray log detects naturally occurring gamma ray radiations such as uranium, thorium, and potassium (Ellis and Singer 2007). In most petroleum geological applications, the gamma ray log is used as a tool to distinguish the potentially productive intervals from probable unproductive shale intervals, and to estimate the shale volume in the shaly reservoir units. Generally, the shaly intervals are characterized by higher level of gamma ray radiation due to presence of clay minerals with high potassium content (such as illite), absorption of thorium by clay minerals, and uranium fixed in organic material. By contrast, sandstone and carbonate rocks, with low clay content, exhibit lower levels of radioactivity. Exceptions can occur, where sandstone (with high K-feldspar, zircon, and mica content) and carbonate ("hot" dolomite) rock readings can raise as high as shaly intervals resulting in more uncertain lithology interpretations. As an enhancement to the natural gamma ray logging, in spectral gamma ray logging, the incoming gamma rays are separated in to a series of energy windows to count the rates of thorium, uranium and potassium. Spectral gamma ray logging is most useful in identifying clay minerals, organic rich rocks, mica sand, "hot" dolomite, natural fractures, and uranium prospecting.

The formation density log measures the bulk density, as the sum of the matrix and pore fluid densities, based on the attenuation of the induced gamma ray radiations. The measured density values can only be used as an indication of lithology when the porosity is insignificant (e.g., in evaporites). Moreover, the density log, when used alone, is not an appropriate tool for identifying most lithology types due to their polymineralic nature and variable porosity values.

The neutron tool emits high energy neutrons into the formation. The neutrons undergo scattering in the formation, lose their energy, and produce high energy gamma rays. The low energy neutrons and the resulting gamma rays can be detected. Significant neutron energy loss occurs in the presence of hydrogen and chlorite. Normally, high concentration of hydrogen is related to the amount and type of pore filling fluids. High amount of dissolved chlorite ions in the drilling mud, mud filtrate or formation fluids can result in high neutron energy loss, which can be incorrectly interpreted as porosity leading to an overestimation of porosity. Moreover, shale contains clay minerals with significant amount of bound water that can result in high proportion of hydrogen despite low porosity. This is referred to as "shale effect", where the apparent porosity readings from the neutron tool is substantially higher than that in carbonate or sandstone rocks. Therefore, a high neutron porosity reading is a partial indicator of shale, and can be distinctive when combined with the gamma ray log. The neutron tools are calibrated to provide accurate porosity values in a limestone formation. The tool readings are, therefore, usually reported in limestone porosity units. This means that the apparent porosity in a limestone formation will be accurate, but porosity readings in any other formation lithology need to be corrected using available correction charts.

The photoelectric absorption log, as a latest generation of density logging tools, measures the photoelectric absorption factor, P_e. The P_e is approximately given by $(Z/10)^{3.6}$, where Z is the average atomic number of the formation, thus, P_e is basically a function of matrix mineralogy. Since fluids have low atomic numbers, they have negligible effects on the P_e log response. Due to lower sensitivity to the pore volume changes compared to the neutron and formation density logs, the P_e log is considered as an excellent indicator of mineralogy. This log is commonly scaled on a range between 0 and 10 barns/electron units. Common mineral reference values are quartz at 1.81, calcite at 5.08, dolomite at 3.14 barns/electron (Bertozzi et al. 1981). Barite is an efficient absorber of gamma rays with P_e of 267 barns/electron. Hence, this tool cannot be used with barite drilling muds. The P_e log is most useful when used in conjunction with other tools such as density and neutron logs.

The sonic log measures the travel time of sound or an elastic wave from a transmitter to a receiver, both mounted on the tool. The wave undergoes dispersion (spreading the energy in time and space) and attenuation (loss of energy through absorption by the formation). The sonic travel time varies for various types of waves (e.g., compressional and shear waves) as different waves travel with different velocities in the rock. The first wave arriving at the receiver is the compressional or P-wave. The velocity of P-wave depends on the elastic properties of the matrix (i.e., composition and microstructure of the matrix), the fluid type and its distribution, and the porosity of the formation. The P-wave slowness is directly proportional to the density of the material. The next wave, usually, is the shear wave or S-wave, which cannot propagate in fluids. Therefore, it provides information on only the solid (rock) material.

The main applications of the sonic log is to provide information to support or calibrate seismic data. However, combined with density and neutron logs, the sonic log can be employed for estimation of porosity and identification of lithology. Other applications of the sonic log include stratigraphic correlation, identification of fractures, compaction, over-pressures and source rocks (Rider 1986).

As mentioned, the density, neutron, and sonic logs respond to porosity variations, and at the same time, they are affected by lithology variations. To address the uncertainties associated with lithology identification, a variety of cross plotting techniques have been developed to obtain lithology by combining the responses of these logs. Although the lithology might include clay and other minerals, these cross-plots often assume three main rock types, i.e., sandstone, limestone, and dolomite. For a detailed description of these techniques, the interested reader is referred to (Poupon et al. 1971; Asquith et al. 2004).

Unique combinations of log responses that reflect specific physical and compositional characteristics of an interval and distinguish it from other intervals is referred to as "electrofacies". Serra and Abbott (1982) considered electrofacies as proxies of lithofacies.

2.3 Fundamentals of CT scanning

X-ray computerized tomography (CT) is a computer enhanced imaging procedure for obtaining photographs of cross-sectional slices of an object. From its first development by Godfrey Hounsfield and Allan Cormack during the early 1970's, X-ray CT scanning has seen unabated improvements that continue to this day. The technology was initially used in the medical science for better identification of internal organs and tissues as well as possible abnormalities. However, it has found wide applications in other sciences including earth science (Vinegar 1986; Ketcham and Carlson 2001).

Whole core CT scanning has a long history in assisting geologists to study extracted cores by providing 2D and 3D information on the texture and internal structure of the reservoir rocks. This technique is non-destructive, and the extracted cores can be scanned using a CT scanner regardless of preservation and whether the core is still in a capped core barrel. CT scanners use an X-ray source that shoots a narrow beam of X-rays at different angles to the object being scanned; either by rotating the source or the object (Goldman 2007; Lopez et al. 2016). As the X-rays pass through the object, they are attenuated, and the resulting reduced X-rays are picked up by the detectors located directly opposite to the source. The attenuation of a mono-energetic beam is described by Beer's law:

$$I = I_0 e^{-\mu h} \quad , \tag{2.1}$$

where I_0 and I are initial and attenuated X-ray intensities, respectively, μ is the attenuation coefficient of the underlying material, and h is length of the X-ray path through the object. The detected X-ray intensities are transmitted to a computer, where the 2D distribution of attenuation values are reconstructed to generate cross-sectional image slices called tomographic images. These images can either be displayed individually, or they can be stacked together to form a 3D representation of the scanned object.

Technically, each voxel (or pixel volume) in the CT images is represented by a gray-level attenuation value that indicates a certain level of attenuation. For low energy levels, X-ray attenuation is dominated by the photoelectric effect, while Compton scattering becomes more dominant for higher energy levels. This leads to different attenuation of a substance at different energy levels. The attenuation coefficient at different energy levels depends on the substance electron density, ρ_e, and atomic number, Z, as described by the following equation proposed by Wellington and Vinegar (1987):

$$\mu = \rho_e \left(k + \tau \frac{Z^{3.8}}{E^{3.2}} \right) \quad . \tag{2.2}$$

Here k is the Klein-Nishina coefficient, E is the energy level, and τ is a constant. The Klein-Nishina coefficient k is assumed to be little influenced by the energy level. The above relationship indicates that at high energy levels the attenuation coefficient will be influenced by the density, while the effect of the atomic number increases with decreasing the energy level. Thus, the effective atomic number (Z_{eff}) and electron density (ρ_e) of an object can be calculated by scanning it at two different energy levels with sufficient energy separation (Wellington et al. 1987).

This technique, known as Dual-Energy CT scanning (DECT) or spectral imaging, is widely used in medical dual-energy X-ray absorptiometry (DEAX) to measure bone mineral density after eliminating the effects of X-ray absorption by soft tissues (Siddiqui and Khamees 2004).

Whole cores scanned at two energy levels can be used to calculate bulk density, ρ_b, of the core material. The bulk density is related to the electron density by the following expression (Gardner and Dumanoir 1980):

$$\rho_b = 1.0704 \times \rho_e - 0.1883 \quad . \tag{2.3}$$

The bulk density calculated from the above equation can be used to compute total porosity, ϕ_t, by the density-porosity equation: (Hartmann and Beaumont 1999)

$$\phi_t = \frac{\rho_{ma} - \rho_b}{\rho_{ma} - \rho_{fl}} \quad , \tag{2.4}$$

where ρ_{ma} and ρ_{fl} are estimated matrix and fluid densities, respectively. Thus, the porosity estimate relies on accurate estimates of matrix and fluid densities.

Chapter 3. Literature review

This chapter provides a detailed review of log- and image-based publications that have utilized AI algorithms to classify lithology and to compute transport properties, and thus relate to or constitute stepping stones for the results obtained in this thesis. The chapter is divided into two distinct sections so to divide the review into a structure that matches the publications produced in this PhD project.

3.1 Application of AI in lithology classification

The application of AI approaches is rapidly increasing in the oil and gas industry. Intelligent exploration, smart drilling, and intelligent production is becoming the direction of the future development in this field (Li et al. 2020). Integration of such advancements can offer strong solutions for effective handling of multi-dimensional data systems. It can reduce human subjectiveness when guiding towards smarter operations with reduced risks. AI is applied across the whole oil and gas value chain; from exploration all the way through production, transportation and end user sales (Ershaghi et al. 2018).

Previously, many publications have employed supervised machine learning techniques to classify lithology based on well logs and core plug measurements. Rogers et al. (1992) determined lithology from selected well logs using a back-propagation neural network (BPNN). Dubois et al. (2007) examined the classical parametric (Bayes' rule) and non-parametric (fuzzy logic, k-nearest neighbor, and feed forward back-propagating artificial neural network) methods in a rock facies classification problem. The authors used a combination of wireline logs responses (natural gamma ray (GR), neutron and density porosity average (PHI), neutron porosity and density porosity difference (N–D), photoelectric effect (PE), and apparent true resistivity (Rta)) and two geologic constraining variables (marine and non-marine indicators) as input to train the considered models. They compared the performance of these models with respect to generalization capabilities on unseen data, and they showed that the artificial neural networks (ANN) outperformed other algorithms, i.e., Bayes' rule, fuzzy logic, and k-nearest neighbor. Al-Anazi and Gates (2010) utilized specific well log responses and core-based measurements (i.e., porosity and permeability) as input variables to integrate support vector machines (SVM) in an interpretation framework consisting of electorfacies identification and permeability prediction in a highly heterogeneous sandstone reservoir. More specifically, they extracted the underlying electorfacies from well logs and core data using extended fuzzy clustering mean analysis (Bezdek et al. 1984). Further, they assessed the potential of SVM to classify electrofacies and predict permeability by comparing its performance against linear discriminant analysis (LDA) and probabilistic neural networks (PNN). The authors showed that the SVM method produced comparable or better results for lithology classification and permeability estimation compared to the neural network methods. Horrocks et al. (2015) investigated the capability of three popular machine learning algorithms, namely the Naïve Bayes classifier, SVM, and ANN, to classify lithology based on highly sampled wireline logs. They trained and evaluated the above algorithms to improve coal classification using a coal-specific performance metric under two architectures: committee (one classifier per well log) and singular (one classifier for all well logs). The authors found that the overall accuracy of all three classifiers increased by application of the committee architecture, and the ANN classifier outperformed the other two classifiers. The capability of SVM was also investigated by Hall (2016), where he used five wireline log measurements and two indicator variables as input to train an SVM to classify nine facies derived from core descriptions. The trained model was then used to assign facies to the wells that had not been described. The prediction results achieved an overall F1-score of 0.43 and showed various misclassification rates within the adjacent facies. Tschannen et al. (2017) employed deep inception network (Szegedy et al. 2015), a modified version of convolutional neural network (CNN), to determine facies utilizing the same data that was used by Hall (2016). These authors achieved a slightly higher averaged F1-score of 0.574 compared to the previous authors. Moreover, they observed model deficiencies in detecting very thin layers, and related it to the lower resolution of wireline logs, used for model training, compared to the core scale lithology descriptions.

The combination of supervised and unsupervised techniques has also been considered in a few publications. As an example, Bize-Forest et al. (2018) combined supervised learning (i.e., SVM Gaussian, ANN, Naïve Bayes, and k-nearest neighbor) and unsupervised gap statistic (Tibshirani et al. 2001) clustering to predict depositional facies from well logs in a complex carbonate reservoir. The supervised methods were simultaneously applied to 1) identify best training method to classify depositional facies given the core-derived depositional facies, and 2) identify the optimum input log combinations. Moreover, they used the gap statistic method to estimate the optimum number of petrophysical clusters. The authors showed that the Naïve Bayes supervised method performed best in clustering the depositional facies and showed the highest training score against the core facies. Dunham et al. (2020) introduced a semisupervised Gaussian mixture model (SSGMM), an algorithm that essentially combines a Naïve Bayes classifier (supervised) and Gaussian mixture models (unsupervised), to improve classification of lithology based on the same set of well log data used by Hall (2016) and Tschannen et al. (2017). The authors compared the performance of this semisupervised model with an existing XGBoost model (i.e., the winner of the SEG machine learning contest (Hall and Hall 2017)), where they demonstrated the semisupervised model outperformed the XGBoost model, but not by a significant margin.

In the majority of the aforementioned publications, well logs have been used as inputs to train models for lithology classification. In addition to these, several publications have considered image-based lithology identification using borehole or core image data, thin sections, or micro-CT images. Linek et al. (2007) introduced a pattern recognition technique to classify lithology based on gray-level coded Formation MicroScanner borehole wall images. More specifically, they extracted Haralick textural features (Haralick et al. 1973) from FMS images and used an appropriate set of features to classify lithology using the Bayes decision rule. The authors also indicated improvements in the classification results by decomposing the images, using wavelet transform, before feature extraction. Jungmann et al. (2011) used different textural features, extracted from borehole wall 2D resistivity images, and a combination of binary LDA classifiers for an automated lithology reconstruction process. More precisely, these authors mapped a multi-class problem into a combination of binary classifiers, and compared the performance of the combined binary classifier with a single multi-class classifier. They found that their combined binary classifier, with certain combinations of textural feature subsets, achieved a higher classification accuracy. Al-Obaidi et al. (2018) used a combination of rock fabric properties extracted from image logs and core measurements to perform an automatic rock classification using a k-means based clustering method. Marmo et al. (2005) trained a multilayer perceptron (MLP) neural network model to classify textures of carbonate rocks using digitalized 8-bit gray-tone images of thin sections. Convolutional neural networks were successfully used by Cheng and Guo (2017) to recognize granularity using colored thin section images as input. Lima et al. (2019) employed transfer learning, i.e., using previously trained models to classify uninterpreted images such as fossil images, slabbed cores, petrographic thin sections and rock and mineral hand samples. Moreover, Anjos et al. (2021) applied deep learning techniques to classify lithology by identifying patterns in pre-salt carbonate micro-CT images.

There exist a few approaches that utilize CT images for facies classification. These approaches employ information content of the CT images through extraction of various features for clustering and classification purposes. Hall and Govert (2016) pre-processed the whole core CT images, extracted statistical features from processed images, and trained a Random Forest classifier to identify bioturbated core intervals. Odi and Nguyen (2018) utilized physical features such as density, porosity and photoelectric effect, extracted from dual energy CT-scans, for supervised (gradient boost, decision tree, neural network, linear regression models, and random forest) and unsupervised (hierarchical clustering) geological facies classifications. In addition, these authors trained a deep learning model to learn the relationship between the CT extracted physical features and existing user-defined geological facies description. These authors did not show concrete results on the application of the trained deep learning model. However, they claimed that their preliminary deep learning approach showed promise in replacing the existing workflows inherent in supervised and unsupervised machine learning. Gonzalez et al. (2019) considered a workflow for an automatic rock classification that combined conventional well logs, whole core CT images, optical core photos, and routine core analysis (RCA) data. In their workflow, rock-fabric related features are first extracted from 2D cross-sectional core CT images and core photos, and then used to determine rock classes by means of a clustering algorithm. Initially, the authors assumed several rock classes, then they optimized this number by iteratively increasing the number of classes and minimizing a permeability-based cost function below a certain threshold. The obtained rock classes were finally used to train an ANN to predict the classes from well log data. In another publication by the same authors, Gonzalez et al. (2020) developed a workflow for detection of image-based rock classes from rock-fabric related features extracted from 3D CT-scan image stacks and slabbed core photos. They, further, used the detected rock classes to improve conventional well log based formation evaluation and permeability estimations.

In this thesis, three approaches have been considered to automatically classify lithofacies based on whole core CT-scan images. In the first approach, first-order statistical features and textural gray-level co-occurrence matrix (GLCM) features are extracted from 2D cross-sectional CT images. Two workflows are considered in this approach. In the first workflow, the extracted features are used to train an SVM to classify lithofacies. In the second workflow, a principal component analysis (PCA) step is added before training. The addition of PCA has two purposes: first, to eliminate collinearity among the features and second, to investigate the amount of information needed to differentiate the analyzed images. Before extracting the statistical features, the images are pre-processed and decomposed using a Haar mother wavelet decomposition scheme (see section 4.1) to enhance the texture and to acquire a set of detail images that are then used to compute the statistical features. The training dataset includes lithological information obtained from core description. The approach is validated using the trained SVM and hybrid (PCA and SVM) classifiers to predict lithofacies in a set of unseen data (**paper I**).

The second approach deals with classification of lithofacies using 2D image slices and CNN. In this approach, 2D images are directly used as input to train a CNN model to learn the relationship between convolution-derived features and expert-derived lithofacies classes (**paper II**).

Finally, the third approach explores the application of the CNN models to classify lithofacies based on the 3D CT images (**paper III**). Thus, this methodology extends the previous approaches working on 2D images into a workflow that uses 3D CT images as direct input to train a CNN model.

3.2 Application of AI in transport property estimation

Porosity and permeability are the most important rock properties in reservoir modeling and formation evaluation processes. These properties and their geometrical distribution can affect the reservoir quality with respect to its viability in storing and producing hydrocarbon volumes.

Among the proposed AI-based methodologies, a high number of publications have introduced workflows for automatic estimation of transport properties based on well log data and artificial neural networks (Wong et al. 1995; Malki et al. 1996; Mohaghegh et al. 1996; Wong et al. 1998; Helle et al. 2001; Al-Bulushi et al. 2009; Saljooghi and Hezarkhani 2014; Elkatatny et al. 2018). These publications have investigated the possibility of predicting porosity, permeability, water saturation, and oil saturation from different combinations of well logs (e.g., gamma ray, bulk density, neutron porosity, sonic, deep induction, and nuclear magnetic relaxation (NMR) logs) using BPNN models.

Ahmadi et al. (2014) evaluated the potential application of fuzzy logic (FL) and least square support vector machine (LSSVM), optimized by genetic algorithm

(GA), to monitor porosity and permeability from petrophysical logs. Their results indicated that implication of GA–LSSVM and GA–FL in prediction can lead to more reliable porosity and permeability prediction. Rafik and Kamel (2017) proposed a two-step approach to predict permeability from well logs, utilizing non-parametric regression in conjunction with multivariate statistical analysis. More precisely, they first classified the well log data into electrofacies types using a combination of PCA, model-based cluster analysis and discriminant analysis. They then employed non-parametric regression techniques such as alternating conditional expectations, generalized additive model and neural networks to predict permeability from well logs within each electrofacies. The performance of various algorithms in predicting porosity and permeability from petrophysical logs were compared by Ahmadi and Chen (2019). More specifically, these authors developed a hybridized predictive model, using LSSVM and hybrid particle swarm optimization and genetic algorithms to predict porosity and permeability. The prediction capability of this hybridized model was then compared with a fuzzy decision tree and an ANN model trained on the same data. The obtained results revealed that the hybridized model could predict porosity and permeability with higher accuracy compared to their other models.

In addition to the aforementioned publications, several publications have evaluated the potential of deep neural networks in predicting porosity and permeability based on pore scale images such as micro-CT and Scanning Electron Microscopy (SEM) images. Srisutthiyakorn (2016) employed multi-layer neural network (MNN) and CNN algorithms to predict permeability from 2D and 3D micro-CT binary segmented images. In his approach, the Minkowski Functionals (porosity, specific surface area, integral of mean curvature, and Euler number) were used as input for the MNN model, while the single/multi-scale 2D and 3D binary images were directly used as input for the CNN model. Their results revealed superior performance of multi-scale 2D CNN model against the MNN and 3D CNN models. Alqahtani et al. (2018) employed a CNN regression model to estimate porosity, coordination number, and average pore size from 2D grayscale micro-CT images, where the ground truth values were computed using pore networks extracted from manually segmented images. Bordignon et al. (2019) proposed a methodology to estimate grain size and porosity distribution using synthetic idealized rocks (sphere packs) and CNN. Tembely et al. (2019) developed a workflow to predict permeability from 3D segmented micro-CT images of carbonate rock samples. More precisely, they extracted specific features from pore network models of the 3D segmented images and trained shallow and deep neural networks to predict permeability, where the ground truth permeability values were derived from lattice-

Boltzmann simulations. Hébert et al. (2020), inspired by Sudakov et al. (2019), used an AutoEncoder to segment 3D grayscale micro-CT images of Berea samples and to estimate porosity values using expert-derived segmented images as training ground truth.

In the aforementioned image-based publications, machine learning and deep learning methods were applied at the pore (micrometer) scale as an extension of Digital Rock Analysis (DRA). In addition to these, Abashkin et al. (2020) considered estimation of petrophysical properties at whole core scale using core photos obtained both in daylight and ultraviolet light. More specifically, the authors generated contrast intensity curves employing horizontal and vertical convolutional filters. These curves together with color descriptive features, as well as laboratory measurements of core plugs (i.e., porosity, permeability, and mineral density) were used as a training dataset for a neural network-based prediction model.

None of the previous publications have considered the applicability of machine learning algorithms for transport property estimation based on whole core CT-scan images. In **paper IV** we propose a methodology to estimate porosity using grayscale 2D core CT slices. More specifically we employ an end-to-end CNN regression scheme to estimate porosity using a training dataset that includes porosity measurements derived from routine core analysis. This methodology provides core scale (millimeter) porosity values compared to the previous publications concentrated on core plug scale (micrometer) estimations.

Chapter 4. Theory

With the purpose of making this thesis sufficiently self-contained, this chapter provides information on the background theory of the applied methods for image processing, feature extraction, lithology, and porosity estimation tasks. The chapter is divided into sections that are then dedicated to describe specific aspects of the tasks mentioned above.

4.1 Wavelet transform

In the first considered methodology for lithology classification (**paper I**), 2D cross-sectional images were decomposed using wavelet transform with the purpose of enhancing the texture of images before feature extraction.

Wavelet transform has been popular in the fields of signal/image processing, denoising and compression, and is defined as a mathematical operation that analyses a particular signal/image in the time and frequency domain by representing it as sum of wavelet functions with various locations and scales (Goupillaud et al. 1984; Porwik and Lisowska 2004).

Wavelet decomposition always starts with a single prototype wavelet $\Psi(t)$, called the mother wavelet, from which the wavelets are derived by scaling and translation. The mother wavelet must be short and oscillatory, meaning that it must have zero average and decay quickly at both ends (Perera et al. 2007).

A function (e.g., a signal) $f(t)$ can be transformed using the mother wavelet. There exist a range of different wavelet transforms, including the Continuous Wavelet Transformation (CWT) and the Discrete Wavelet Transformation (DWT).

The CWT of a bounded function $f(t)$ is derived as (Lee 2000)

$$CWT_\Psi f(a,b) = (f, \Psi_{a,b}) = \int_{-\infty}^{+\infty} f(t)\Psi_{a,b}(t)\,dt \quad , \tag{4.1}$$

where

$$\Psi_{a,b}(t) = |a|^{-\frac{1}{2}}\Psi\left(\frac{t-b}{a}\right) \quad . \tag{4.2}$$

As mentioned above, $\Psi(t)$ is the mother wavelet, a is the scale factor, and b is the translation factor. Thus, the CWT transforms the one-dimensional function $f(t)$ into a two-dimensional function $CWT_{\Psi}f(a,b)$, where the two dimensions represent scale and translation.

As indicated by the name, the discrete wavelet transformation is not continuous, and the mother wavelet is translated into discrete steps by selecting

$$a = a_o^m \quad ; \quad b = nb_o a_o^m \quad , \tag{4.3}$$

where a_o(>1) and b_o(>0) are fixed real values, and m and n are positive integers (Lee 2000). The discretized mother wavelet is then defined as

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{a_o^m}}\Psi\left(\frac{t-nb_o a_o^m}{a_o^m}\right) \quad . \tag{4.4}$$

The corresponding DWT becomes

$$DWT_{\Psi}f(m,n) = (f,\Psi_{m,n}) = \int_{-\infty}^{+\infty} f(t)\Psi_{m,n}(t)\,dt \quad . \tag{4.5}$$

DWT decomposes a function $f(t)$ into several scales representing different frequency bands with a bandwidth that increases linearly with frequency (Lee 2000). This is based on Mallat tree algorithm for wavelet decomposition (Mallat 1989). As shown in Figure 4.1, Mallat tree algorithm for wavelet decomposition consists of successive filtering of the signal using a series of low-pass L(n) filters and their dual high-pass H(n) filters, as well as down sampling by a factor of 2 (shown with downward arrow behind 2 in Figure 4.1). A low-pass filter is the basis for most smoothing methods. Applying a low-pass filter tends to preserve low frequency information and remove high frequency information. However, a high-pass filter is the basis for most sharpening methods and applying it retains high frequency information. The type of the selected mother wavelet determines the filter coefficients. The outputs, i.e., D_1, D_2, and D_3 are called the detail wavelet coefficients, while the output of the last low-pass filter is referred to as the approximation wavelet coefficient.

One type of wavelet decomposition of an image is carried out by splitting the image into four subbands, namely HH, HL, LH, and LL subbands (Figure 4.2), where the

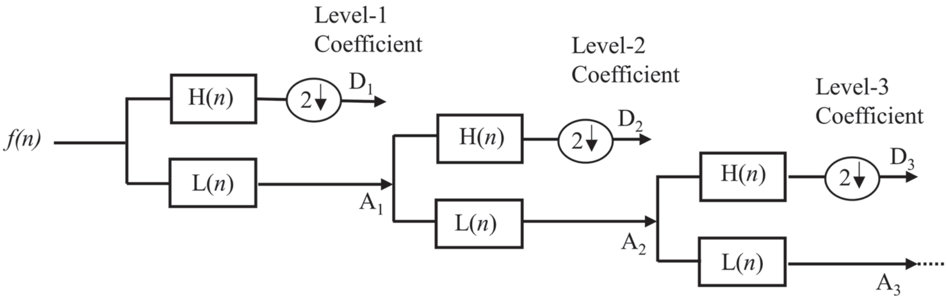


Figure 4.1: Principle of Mallat tree algorithm for wavelet decomposition (Mallat 1989).

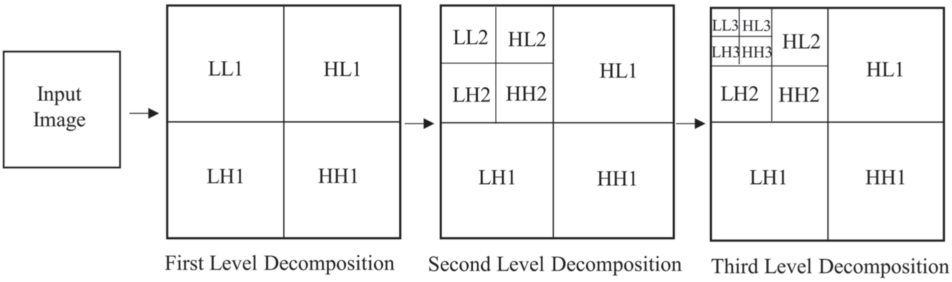


Figure 4.2: Three level wavelet decomposition.

letters H and L represent high- and low-pass filters, respectively. The HH subband represents the diagonal detail of the image. The HL and LH subbands give the horizontal and vertical detail of the image that are acquired by applying a low-pass filter in one direction and a high-pass filter in the other direction. The LL subband is the approximation image, which preserves the low frequency components. By repeating the same process on the approximation component, a higher level of decomposition is obtained. Subbands after three levels of wavelet decomposition are shown in Figure 4.2.

In this thesis, 2D cross-sectional images were decomposed to enhance texture before rock-fabric-feature-extraction step (**paper I**). The images were decomposed using Haar mother wavelets (Porwik and Lisowska 2004). The Haar wavelets consist of a sequence of rescaled "square-shaped" Haar wavelet mother functions defined as:

$$\Psi_H(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \\ 0 & t < 0 \text{ and } t \ge 1 \end{cases} . \tag{4.6}$$

We used three levels of decomposition. More specifically, the vertical detail images at three levels of decomposition were used to enhance the rock texture and to differentiate features and lithofacies vertically.

4.2 Feature extraction

As mentioned before, in this thesis one of the considered approaches for lithofacies classification of the 2D cross-sectional images was to represent these images by features. More precisely, first- and second-order statistical features were extracted from 2D cross-sectional CT images. These features were later on used as input to an SVM model to classify lithology.

4.2.1 First-order statistical features

To capture the information about distribution of the individual pixel values and gray levels in the images, four first-order statistical features, i.e., mean, variance, kurtosis, and skewness, can be extracted from images using the standard definitions:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad , \tag{4.7}$$

$$\sigma^2(x_1,...,x_N) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \quad , \tag{4.8}$$

$$S(x_1,...,x_N) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_i - \bar{x}}{\sigma}\right]^3 \quad , \tag{4.9}$$

$$K(x_1,...,x_N) = \left\{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_i - \bar{x}}{\sigma}\right]^4\right\} - 3 \quad , \tag{4.10}$$

where, x, $\bar{x}$, σ^2, S, and K are the gray-level attenuation values, mean, variance, skewness, and kurtosis, respectively, while N is the total number of pixels.

4.2.2 GLCM

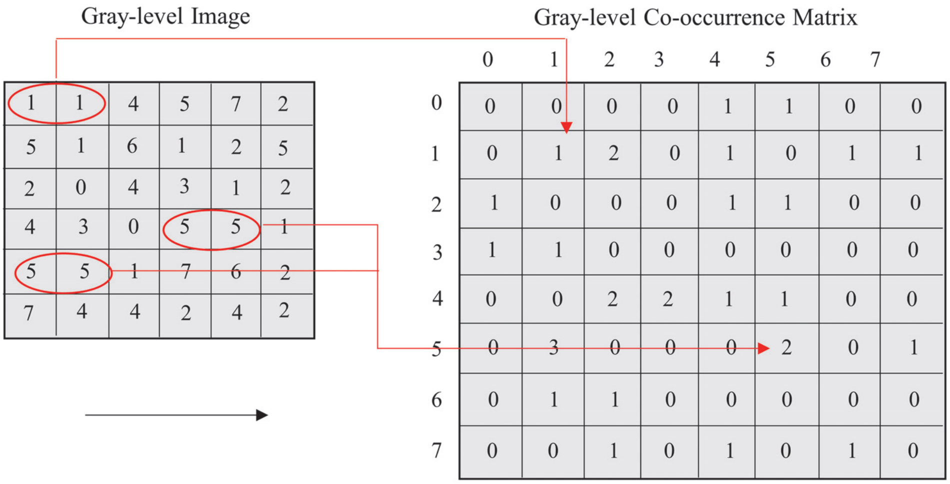
First-order statistics do not capture the spatial interactions and relative position of the pixel values. This type of information can be detected by second-order statistical features. For this purpose, textural statistical features termed gray-level co-occurrence matrix (GLCM) can be extracted from wavelet transformed detail images.

GLCM, also known as gray-level spatial dependence matrix, is a statistical method that considers spatial relationships between pixels in an image. In other words, the GLCM method represents the texture of an image by calculating the occurrence of pairs of pixels with specific values in a specified offset distance and direction (Haralick et al. 1973). The spatial relationships of the pixel pairs are stored in a co-occurrence matrix, denoted as $G(i, j)$, whose values represent the relative frequency with which two pixels, one with intensity *i* and the other with intensity *j*, occur within a given offset distance and direction. GLCM computation is outlined through a schematic representation example in Figure 4.3, where the matrix is calculated on a simplified grayscale image with one pixel offset in the west-east direction. GLCM features are often calculated along horizontal, vertical, and two diagonal orientations. The GLCM computation along each of these orientations is accomplished by addition of GLCM in one direction and the transpose of the GLCM in the opposite direction (e.g., GLCM west-east and transpose of GLCM east-west). Once the GLCM is computed, it is normalized by the number of pixels in the image. Note that the averaged GLCM features along all orientations are used as the final computed features.

In the current study, since we are dealing with non-stationary images with more than one texture, the GLCM features are calculated within a small region of the image using a predefined window. This means placing a rectangular window, covering the width of the image, at the center of each pixel and then computing the GLCM within this mask. Then, the textural feature value is calculated from the acquired GLCM, and that value is assigned to the center pixel of the window. The window is then moved by one pixel, and the process is repeated until all the pixels of the image are covered. This process results in a new image that represents the desired textural feature.

Generally, up to 14 textural features can be extracted from grayscale images. In this thesis we will only consider three of these features; contrast D, energy E, and correlation ρ. The following equations define these features.

$$D = \sum_{i=1}^{P} \sum_{j=1}^{P} (i - j)^2 \times G(i, j) \quad , \tag{4.11}$$



Gray-level Image

1	1	4	5	7	2
5	1	6	1	2	5
2	0	4	3	1	2
4	3	0	5	5	1
5	5	1	7	6	2
7	4	4	2	4	2

Gray-level Co-occurrence Matrix

	0	1	2	3	4	5	6	7
0	0	0	0	0	1	1	0	0
1	0	1	2	0	1	0	1	1
2	1	0	0	0	1	1	0	0
3	1	1	0	0	0	0	0	0
4	0	0	2	2	1	1	0	0
5	0	3	0	0	0	2	0	1
6	0	1	1	0	0	0	0	0
7	0	0	1	0	1	0	1	0

Figure 4.3: GLCM calculation with one pixel offset along the west-east direction. As an example, in the gray-level image along west-east direction, co-occurrence of pair of pixels with value of 5 is 2, which is represented in the resulting GLCM. Note that the GLCM features are computed and averaged along four main orientations. For simplification purposes, here we only show the west-east direction used in the case of Horizontal GLCM computation. This illustration is from **paper I**.

$$E = \sum_{i=1}^{P}\sum_{j=1}^{P} G(i,j)^2 \quad , \tag{4.12}$$

$$\rho = \sum_{i=1}^{P}\sum_{j=1}^{P} G(i,j)\frac{(i-\mu_{G)}(j-\mu_G)}{\sigma_G^2} \quad . \tag{4.13}$$

In the equations above $G(i,j)$ is the matrix of relative frequencies with which two neighbor pixels with intensities of i, and j, occur in the image. P is the total number of grayscale values of the GLCM, μ_G is the GLCM mean, and σ_G^2 is the variance of the intensity of all reference pixels in the relationships that contributed to the GLCM.

In **paper I** a total of 13 features were computed and used to represent a set of 2D cross-sectional images; four first-order statistical features (mean, variance, skewness, and kurtosis), and three GLCM features (contrast, energy, and correlation) for each of the three levels of decomposition.

4.3 Principal component analysis (PCA)

PCA (Pearson 1901) is a multivariate data analysis technique that is used in a wide range of applications, including data exploration, visualisation of underlying patterns within correlated datasets, decorrelation, detection of outliers, data compression, feature reduction, and more. One of the applications of PCA is to reduce dimensionality of a dataset consisting of a large number of interrelated variables, while preserving as much variance as possible. This is achieved by a linear transformation into a new coordinate system given by the orthogonal principal components, for which the original data points are projected onto this new coordinate system. The principal components are uncorrelated, and the first principal component explains most of the variation in the original dataset, whereas the second highest variance is captured by the second principal component, and so on.

As an illustration of PCA, consider a mean-centered two-dimensional vector space x_1 and x_2 with observations, where each observation corresponds to a single point in the vector space, as exemplified in Figure 4.4. To find a basis vector which represents the statistical variation of the data, PCA fits a single straight line approximating to the observations in the least-square sense, i.e., minimizing the sum of distances between data points and the line. The first principal component is the vector of unit length in direction of the line to which the original data points have been projected, denoted as z_1 in Figure 4.4. This principal component explains the highest variation in the original data points. The second principal component explains the second highest variation, and it is perpendicular to the first principal component minimizing the distances from data points to the corresponding straight line (z_2 in Figure 4.4). This process is repeated for higher dimensional observation spaces.

A PCA can be computed employing the eigenvalue decomposition (EVD) method. Basically, the EVD is a factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors. A non-zero vector V of dimension N is an eigenvector of a square $N \times N$ matrix A if it satisfies the linear equation:

$$AV = \lambda V \quad , \tag{4.14}$$

where λ is the eigenvalue corresponding to V. Geometrically, the eigenvectors are the vectors that the linear transformation A stretches/squishes, and the eigenvalues are the factors by which the eigenvectors are stretched/squished (Alexandris et al. 2017).

Consider a dataset consisting of P observations, each with D variables, where

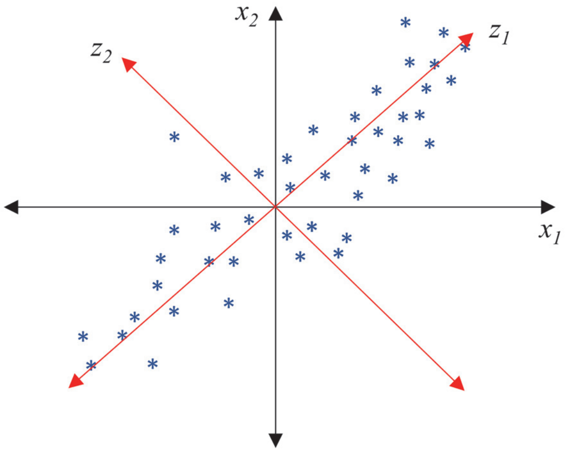


Figure 4.4: Principal component analysis of a mean-centered two-dimensional vector space.

usually $P \gg D$. PCA can be used to reduce the dimensionality of the data by representing each observation with only F variables, where $1 \leq F < D$. The EVD-based principal component transformation is performed in different steps as described below.

The first step is to arrange the observations into row vectors in a data matrix X of shape $P \times D$, where each row represents a single observation with D variables.

The second step is to mean-center the data by computing the row-wise mean, i.e., the mean $\bar{x}_j$ of all rows j, as

$$\bar{x}_j = \frac{1}{P} \sum_{i=1}^{P} X_{ij}, \quad \text{where} \;\; j = 1, ..., D \quad . \tag{4.15}$$

We then have the vector of row means $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D)$, which gives the mean matrix as

$$\bar{X} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{x} \quad . \tag{4.16}$$

Subtracting $\bar{X}$ from X we get the mean-centered data B:

$$B = X - \bar{X} \quad . \tag{4.17}$$

Once the data is mean-centered (zero mean), the next step is to compute the covariance matrix of the rows of B as

$$C = \frac{1}{P-1} B^{\top} B \quad . \tag{4.18}$$

The principal components can then be obtained by computing the eigen decomposition of the covariance matrix C :

$$CV = \lambda V \quad , \tag{4.19}$$

where, as stated above, V and λ are eigenvectors and eigenvalues of the covariance matrix, respectively. The eigenvectors indicate the variation in the dataset, and they can be sorted with respect to their decreasing order of eigenvalues. Choosing the F eigenvectors with the highest corresponding eigenvalues, then these F eigenvectors will capture most of the variation in the original data points given only F out of D free variables.

In this thesis, the PCA analysis was performed with two purposes in **paper I**: first, to reduce dimensionality and eliminate the collinearity among the extracted features; and second, to investigate the amount of information needed to distinguish lithofacies classes.

4.4 Employed machine learning algorithms

This thesis relies heavily on two specific machine learning algorithms, namely support vector machines and convolutional neural networks. These approaches were applied for lithofacies classification (**paper I, II, and III**) and porosity estimation (IV), and will be briefly described below.

4.4.1 Support vector machines (SVM)

SVM, developed in the framework of statistical learning theory (Cortes and Vapnik 1995; Vapnik 2000), has found various applications as diverse as time series prediction to face recognition and medical diagnosis. A brief overview of the statistical learning theory (STL) and the theoretical foundations of SVM are provided below.

The problem of supervised learning, in STL, is formulated as follows. Let us assume a set of l training data $\{(x_1, y_1), ..., (x_l, y_l)\}$ in $\mathbf{R^n} \times \mathbf{R}$ sampled with unknown probability of $P(x, y)$, and a loss function $L(y, f(x))$, where L measures the error, for a given x, between predicted, $f(x)$, and the actual value y. The task is to find a function f (or train a model) that minimizes the expected error on new data (Evgeniou and Pontil 1999):

$$\int L(y, f(x))P(x, y)dxdy \quad . \tag{4.20}$$

Since the $P(x, y)$ is unknown, we need to use some principle of inductive learning to infer a function that minimizes the expected error from l available training examples. The principle is called Empirical Risk Minimization (ERM), which is used over a set of possible functions that map the input space to the output space. These functions are also called hypothesis space, H. The main idea behind using the ERM is to approximate the expected error (or risk) using the empirical mean over the set of training data points:

$$\frac{1}{l}\sum_{i=1}^{l} L(y_i, f(x_i)) \quad , \tag{4.21}$$

where f is restricted to be in the hypothesis space. The relationship between the empirical and expected errors depends on the capacity or complexity of H. The more complex H is, the larger the distance between the empirical and expected errors (i.e., overfitting). Therefore, to choose the optimal solution to the learning problem, it is important to measure the complexity of a hypothesis space.

A standard quantity in STL, used to determine the complexity of a hypothesis space, is VC (Vapnik–Chervonenk) dimension (Vapnik and Chervonenkis 1971; Vapnik 1999; Vayatis and Azencott 1999), a combinatorial quantity that determines the capacity of the set of functions to shatter a set of points. The function $f(x)$ can shatter points $(x_1, ..., x_l)$ if for all $(y_1, ..., y_l)$, $f(x)$ achieves zero error on training data $(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)$. The VC dimension of the function $f(x)$ is the maximum number of points that can be arranged so that $f(x)$ can shatter them.

If h is the VC dimension of H, then we can show that with probability of $1 - \eta$ (where η is a small number) the minimum expected error (or test error), L, will be upper bounded by the minimum empirical error (train error), L_{emp} as (Vapnik 2000):

$$L \leq L_{emp} + 4\sqrt{2}\sqrt{\frac{h(1+log(\frac{2l}{h})) - log(\frac{\eta}{4})}{l}} \quad , \tag{4.22}$$

independent of the probability distribution of data, $P(x, y)$. The right hand side of Eq. (4.22) is called "risk bound", and the second term on the right hand side is called the "VC confidence" (Burges 1998).

SVM realizes the ideas outlined above to find an optimal hyperplane to separate data points in an n-dimensional space.

The simplest formulation of SVM is the linear case, where the hyperplane lies on the same space as the input data. Assume a training dataset consisting of l points of form $(x_1, y_1), ..., (x_l, y_l)$, where y_i indicate the class of point x_i, and it is either 1 or -1. Then, the points on the hyperplane satisfy:

$$w^\top x - b = 0 \quad , \tag{4.23}$$

where w and b are the parameters of the hyperplane, with w normal to the hyperplane. The parameter $\frac{b}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin along the normal vector w.

In a linearly separable dataset, we can find two parallel hyperplanes that separate the two classes in such a way that the distance between them is as large as possible. The region between these hyperplanes is called the "margin", and the hyperplane that lies halfway between these two hyperplanes is called the "maximum margin hyperplane". The two hyperplanes, using a normalized dataset, can be described by:

$$w^\top x - b = 1 \quad , \tag{4.24}$$

and

$$w^\top x - b = -1 \quad . \tag{4.25}$$

Geometrically, the distance between these hyperplanes is $\frac{2}{\|w\|}$. Therefore, to maximize this distance, $\|w\|$ should be minimized. In addition, to prevent the data points from falling into the margin (i.e., incorrect classification), they need to satisfy the following constraints:

$$w^\top x_i - b \geq 1, \text{ if } y_i = 1 \quad , \tag{4.26}$$

or

$$w^\top x_i - b \leq -1, \text{ if } y_i = -1 \quad . \tag{4.27}$$

This can be combined into one set of inequalities as:

$$y_i(w^\top x_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq l \quad . \tag{4.28}$$

Based on this, the expected solution for a typical two-dimensional case should have the form shown in Figure 4.5. The training data points for which the inequality in Eq. (4.28) holds (i.e., the data points lying on one of the hyperplanes) determine the optimal hyperplane (solution), and they are called "support vectors". The removal of support vectors would change the solution found. This type of SVM is called "hard margin SVM" since strict constraints need to be satisfied to correctly classify the data points.

If one enforces perfect classifications of the training data when the underlying classes overlap, one risks getting hyperplanes that are very complicated, and this can affect the generalization capability of the trained model. Therefore, SVM allows for misclassifications during the training phase by using a "soft margin" technique.

The above mentioned SVM method works for linear classification cases. To extend this method to the non-linear cases, Boser et al. (1992) suggested a methodology that applies the so called kernel trick (Aizerman 1964) to the maximum-margin hyperplanes. Employing this technique, the data points are mapped into a higher dimensional feature space, where the data points are linearly separable. Although, the classifier is a hyperplane in the higher dimensional feature space, it might be nonlinear in the original input space. The math behind the kernel trick is not discussed here, and the interested reader is referred to the literature.

There are different kinds of kernels such as linear, nonlinear, polynomial, Gaussian radial basis function (RBF), and sigmoid functions. The choice of the kernel function can affect the results of the SVM classification. From the mentioned kernels, radial basis functions have been among the most popular kernel functions, and they have been shown to perform better than a sigmoid kernel (Lin and Lin 2003). The Gaussian radial basis function has the following format on two data

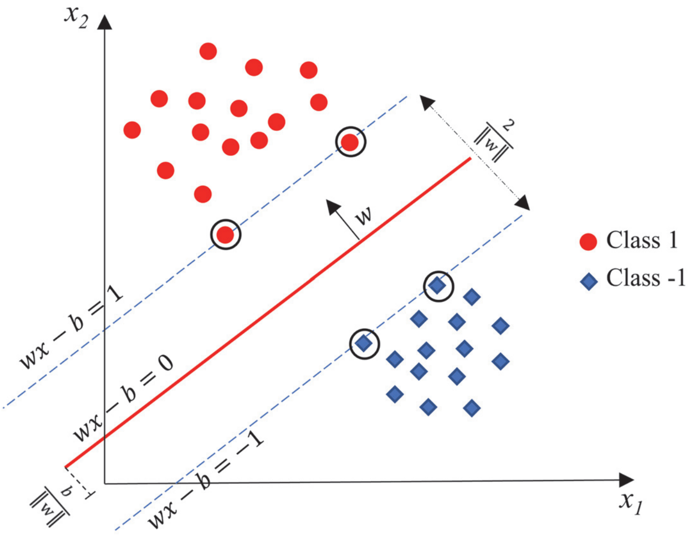


Figure 4.5: The SVM solution for a two-dimensional linearly separable case. The support vectors are circled (this figure is modified from Burges (1998)).

points of x_i and x_j, represented as feature vectors in some input space:

$$F(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \quad , \tag{4.29}$$

where $\|x_i - x_j\|$ is the squared Euclidean distance between the two feature vectors, and γ is the Gaussian kernel parameter.

4.4.2 Convolutional neural networks (CNN)

CNNs, initially proposed by LeCun et al. (1989), have proven to be specifically effective in the fields of image recognition, voice recognition, and classification. In general, neural networks are inspired by the way the human brain works. This class of algorithms learns the relevant features directly from the input training data, so there is no need for manual feature extraction by a subject matter expert. Most of the modern CNN architectures consist of alternating convolutional and pooling layers followed by fully connected layers. The convolutional and pooling layers deal with feature extraction, while the fully connected layers map these extracted features into the final output. Details of the CNN architecture is provided below.

Convolutional layers

A convolution is a mathematical linear operation that can be used for feature extraction purposes. More precisely, consider a digital image as represented by a 2D array of numbers (pixels). A set of optimizable convolutional filters (called convolutional kernels) are superposed in each position of the image. The area of the image covered by the kernel is called the receptive field. An element-wise multiplication between the elements of the kernel and the receptive field in the input image is performed, and the product results are summed up and stored in the corresponding position in the feature map, i.e., the output of the convolutional layer. Once the convolution operation is computed and stored for that specific location, the kernel is then moved either horizontally or vertically by an offset called "stride". This process is repeated until the entire image is covered and the resulting feature map is completely populated (see Figure 4.6).

Different kernels act as different feature extractors: low level features are extracted in the first convolutional layer and, as the output of one layer feeds into the next layer, higher level and more complex features are extracted hierarchically (Yamashita et al. 2018). Convolutional layers are locally connected, whereas in the classic neural networks each neuron is fully connected to the neurons in the other layers.

As mentioned, convolution is a linear operation. In order to introduce non-linearity, one may let the outputs of the convolution operations pass through an opportunely designed activation function. The most common activation function is the rectified linear unit (ReLU); the advantage of using this specific function is that it allows fast and effective convergence during the training process. ReLU is defined as:

$$g(x) = \max(0, x) \quad . \tag{4.30}$$

The operation in the convolutional network can be written as (modified from Anjos et al. (2021)):

$$o_i = g((o_{i-1} * w_i) + b_i) \quad , \tag{4.31}$$

where o_i is the output of the i^{th} layer, o_{i-1} is the output of the previous layer, $*$ is the convolution operation, w_i is the kernel weights of the i^{th} layer, b_i is the bias vector of the i^{th} layer, and g is the non-linear activation function.

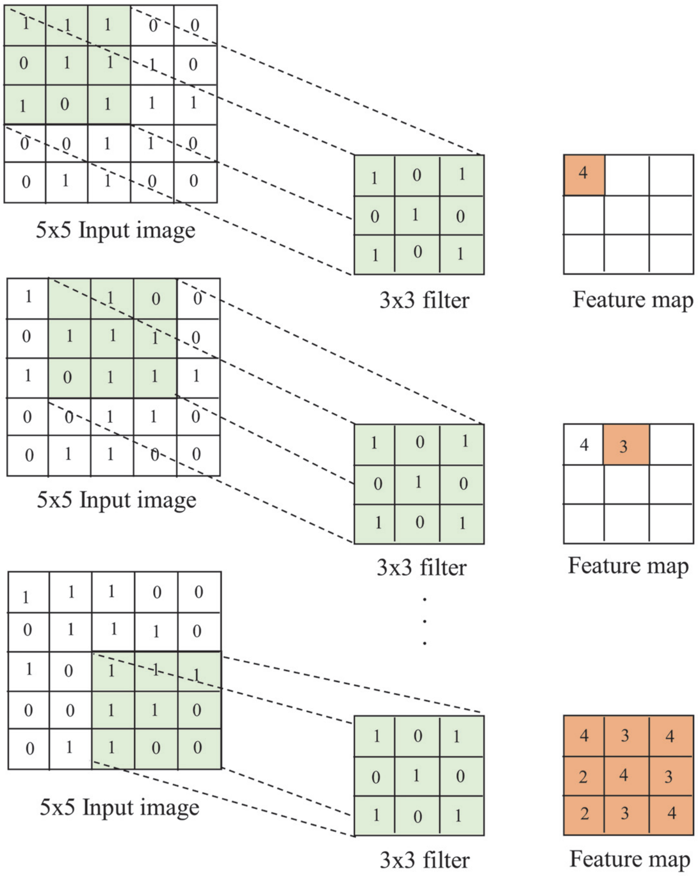


Figure 4.6: An example of convolution operation with a stride of 1 and no padding. The input image is represented by a 2D array and a filter of size 3x3 is applied on the receptive field. An element-wise multiplication is performed and the product results are summed up and stored in the feature map. This figure is a reworked version of Figure 3 in (Yamashita et al. 2018).

Pooling layers

After applying the activation functions, a pooling layer is added. The feature maps derived by convolutional operation record the exact position of the existing features. Therefore, minor spatial changes in the input image yields a different feature map. The pooling layer is applied with two main purposes. First, to make the extracted feature maps less sensitive to local translations and spatial variations in the input image, such as edges, angles, feature positions etc. Second, to reduce the spatial dimension of the feature maps, thereby reducing the computational time associated with model training. Pooling is analogous to convolutional operation, where a window is sliding over the input image. However, it executes a selection of elements inside the pooling window without trainable parameters. The most common pooling operations are maximum pooling and average pooling. Maximum pooling picks the maximum value inside the window and discards the others, while average pooling computes the average of values inside the pooling window.

Fully connected layers

The pooled feature map output of the convolutional section is flattened and fed into fully connected layers that map the extracted features into the final output layer for class predictions. The fully connected layers are basically a classical neural network, where the input nodes are fully connected to the output nodes by learnable weights. Here, non-linearity can also be introduced by adding an activation function such as ReLU. Mathematically, these layers can be defined as (Anjos et al. 2021):

$$z_i = g(z_{i-1} w_i + b_i) \quad , \tag{4.32}$$

where z_i is the output vector, z_{i-1} is the output of the previous layer in the fully connected network, w_i is the weights tensor of the i^{th} layer, b_i is the bias of the i^{th} layer and g is the activation function. For the first layer, z_{i-1} is the flattened version of the last convolution layer, i.e., o_{i-1}. Note that the activation function applied to the final fully connected layer is normally selected based on the type of the task, e.g., classification or regression. The softmax function is a common activation function for multi-class classification, which returns the probability distribution of the predicted classes.

Training a CNN model

The CNN training is a process by which the kernel weights in the convolutional layers and weights in the fully connected layers are adjusted in such a way that the difference between the predicted labels and the actual labels is minimized. Training is commonly performed by a forward and back-propagation process throughout the entire network using a gradient descent optimization algorithm and a loss function. The loss function computes the difference between the output predictions, computed through forward propagation, and the actual label. The network performance is evaluated using the loss function. Cross entropy (L_c) is typically used as the loss function for multiclass classification tasks, whereas the mean squared error (MSE) or mean absolute error (MAE) is typically used for prediction of continuous values (i.e., in regression problems). These are formulated as:

$$L_c = -\frac{1}{l}\sum_{i=1}^{l} y_i \log\left(p\left(\widehat{y}_i\right)\right), \tag{4.33}$$

$$MSE = \frac{1}{l}\sum_{i=1}^{l}(y_i - \widehat{y}_i)^2, \tag{4.34}$$

$$MAE = \frac{1}{l}\sum_{i=1}^{l}|y_i - \widehat{y}_i|, \tag{4.35}$$

where y_i and $\widehat{y}_i$ are respectively the true and predicted labels of the i^{th} sample, p is the probability, while l is the total number of training samples.

During training, the learnable parameters are updated iteratively using a gradient descent optimization algorithm that seeks to minimize the loss function. Basically, the partial derivative of the loss function with respect to each learnable parameter is first calculated; once the whole loss function gradient is computed, the learnable parameters are updated using (Yamashita et al. 2018):

$$w^+ = w - \alpha\frac{dL}{dw} \tag{4.36}$$

where w refers to each learnable parameter with w^+ being the updated value, α stands for learning rate, and L is the loss function. The learning rate is an important hyperparameter that determines how fast the learnable parameter (weight) should move in the direction of the gradient. Note that finding the optimal learning rate during training is crucial for neural networks, since the training process may not converge when using a too high learning rate.

To avoid this issue, it is common to employ various types of optimizers so to search the optimum parameters using a pool of different gradient descents strategies, among which one then choose the best. Empirical comparisons of different optimizers can be found in (Choi et al. 2019). Examples of descent methods are stochastic, batch and mini-batch gradient descents. These methods vary in terms of the number of samples used to compute the error between the actual and predicted labels. For an overview of the gradient descent algorithms, the interested reader is referred to (Ruder 2016).

Generally, in addition to the learnable parameters, there exists a second type of parameters, referred to as hyperparameters. These are not learnable by network, and need to be set by the user. Examples of hyperparameters are the learning rate, type of optimizer, the number of convolutional layers, the number of kernels in the convolutional layers, and number of hidden layers and neurons in the fully connected layers. The performance of a CNN model is highly dependent on the right choice of the hyperparameters. The process of adjusting the hyperparameters is called hyperparameter tuning.

Overfitting

Overfitting is one of the most common problems in machine learning, and it refers to a phenomenon when the model memorizes irrelevant noise instead of learning the relevant features in the training data. Therefore, it fails to generalize on the new data. A common way of recognizing overfitting is to monitor the loss and accuracy on the training data and a hold-out set of data, referred to as validation set. If the model performs worse on the validation set compared to the training set, then the model has most probably been overfitted to the training data.

Several solutions have been proposed to minimize the overfitting in the CNN models. Obtaining more training data is the first and the best proposed solution, because the CNN models trained on large amount of data are proven to generalize better. Other solutions include regularization with dropout, weight decay, batch normalization, data augmentation, and reducing architectural complexity (Yamashita et al. 2018). Dropout is a regularization technique, where randomly selected neurons are discarded during training (i.e., they are temporarily removed from the network together with their incoming and outgoing connections). This way the model becomes less sensitive to specific weights in the network (Hinton et al. 2012; Srivastava et al. 2014). Weight decay, also called $L2$ regularization, penalizes the model complexity by adding a penalty term to the loss function, so that the weights can only take small values (Hanson and Pratt 1988). In batch normalization, the output of a convolutional layer is adaptively normalized before being used in the next one. This technique is known to also have a regularization effect, and it is empirically known to typically speedup the network training, plus make it less sensitive to the initialization point (Ioffe and Szegedy 2015). Data augmentation is one of the most popular techniques, where the main idea is to expand the training dataset by applying transformations. This technique is commonly utilized in image processing, where images are transformed using operations like rotation, pixel shifts, flipping, and random cropping.

Despite of the mentioned efforts, the model might overfit to the validation set rather than to the training data due to information leakage during the hyperparameter tuning and model selection processes. Therefore, testing the model performance on a separate unseen test set is crucial for evaluating the model generalizability.

Chapter 5. Material

The employed material in this thesis includes whole core CT-scan images, manual core description, and routine core analysis data. This chapter will present the material, with one section for each of these three data types.

5.1 Whole core CT-scan data

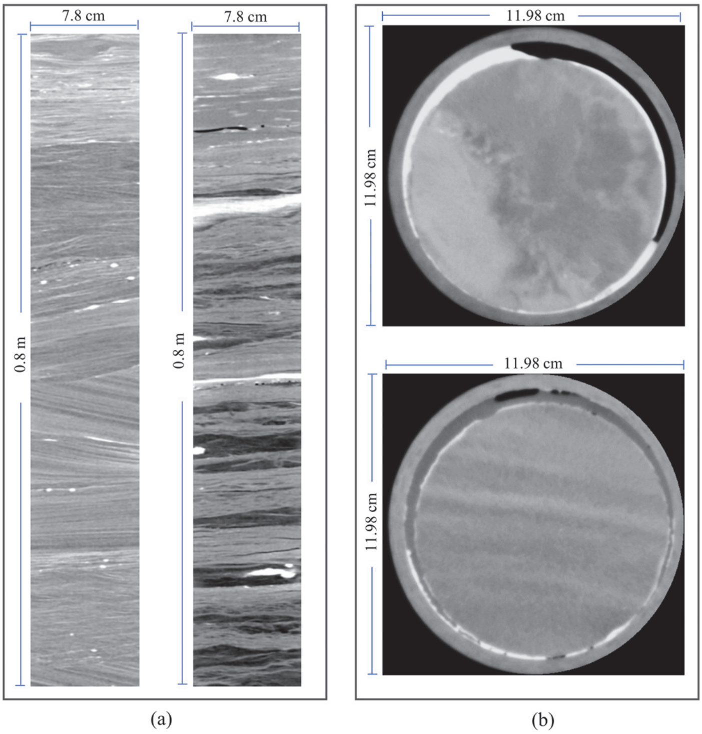
This study utilizes whole core CT-scan images from two wells on the Norwegian continental shelf. Two types of CT images have been used in the proposed methods including, 2D cross-sectional images and 2D image slices. Examples of these two types of images are shown in Figure 5.1. **Paper I** uses 2D cross-sectional images, while the other three publications (i.e., **paper II**, **paper III**, and **paper IV**) utilize 2D image slices. Cross-sectional images were provided in grayscale 8-bit format, while the image slices were provided in 16bit unsigned DICOM format (a standard format developed for medical images (Mustra et al. 2008)) with horizontal and vertical resolution of approximately 0.234 and 0.45 millimeters, respectively. These images require a step of pre-processing, (i.e., cropping, intensity adjustment, and artefact removal) before being used as input for the considered algorithms. Image pre-processing procedures will be explained in detail in section 6.1.

5.2 Lithology and manual core description

As mentioned, CT-scans of two wells are considered in this study. The first well penetrates four main formations, denoted as Fm.1, Fm.2, Fm.3, and Fm.4 in Figure 5.2. Formation 1 consists of very fine-grained argillaceous sandstones and cemented sandstones; Formation 2 consists of successive layers of mudstones and fine-grained sandstones; Formation 3 consists of granule rich medium-grained sandstones and spiculites (a biogenic rock composed of sponge silica spicules); Formation 4 comprises mud and calcite rich marlstones.

The overall lithology in this well is divided into 20 lithofacies by manual core description. The abbreviated labels together with a short core description of each lithofacies class are presented in Table 5.1. Examples of 2D cross-sectional images of these lithofacies classes are also shown in Figure 5.3.

The second well, however, penetrates only two of the previously mentioned formations, i.e., Formation 3 and Formation 4 as shown in Figure 5.4. Here, the lithology of Formation 4 is very similar to the first well, consisting of mud and caliche rich marlstones. However, Formation 3 is quite different than the one in the first well, as it contains also intervals with more coarse-grained lithofacies.



(a) (b)

Figure 5.1: Examples of 2D cross-sectional (a) and image slices (b) employed in this thesis.

5.3 Routine core analysis data

Porosity and permeability data from RCA measurements have also been utilized in this thesis. The porosity-permeability cross-plots, for both wells, are shown

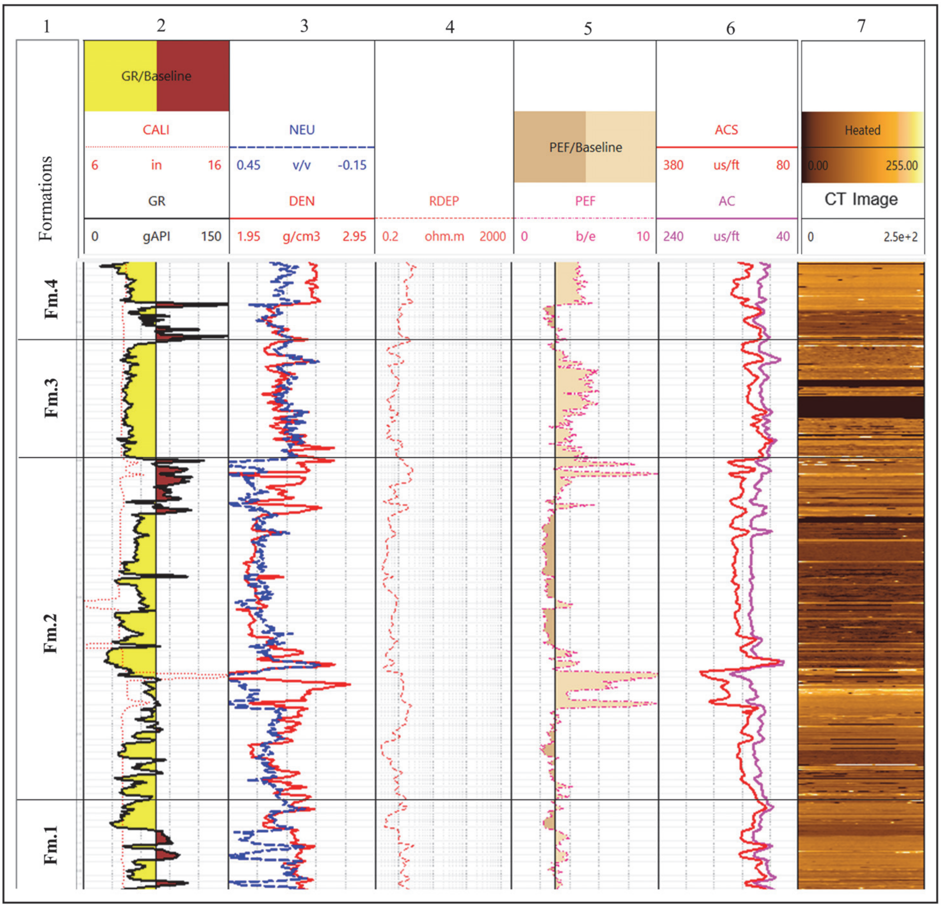


Figure 5.2: Well log data and 2D cross section of the core CT image showing 142 meters of the first well. Log tracks from left to right: track 1: Formations, track 2: Caliper (CALI) and Gamma ray (GR), track 3: Density (DEN) and Neutron (NEU), track 4: Deep resistivity (RDEP), track 5: Photoelectric factor (PEF), track 6: Compressional wave slowness (AC) and shear wave slowness (ACS), track 7: 2D cross section of whole core CT-scan.

in Figure 5.5. The porosity-permeability relationships were utilized to, quantitatively, identify similar lithofacies classes in **paper I** and **paper II**. Moreover, RCA-derived porosity values were used as ground truth for porosity estimation in **paper IV**.

As shown in Figure 5.5, the first well contains a whole range of porosity values, from approximately 0.03 up to 0.40, while permeability values range from 0.01

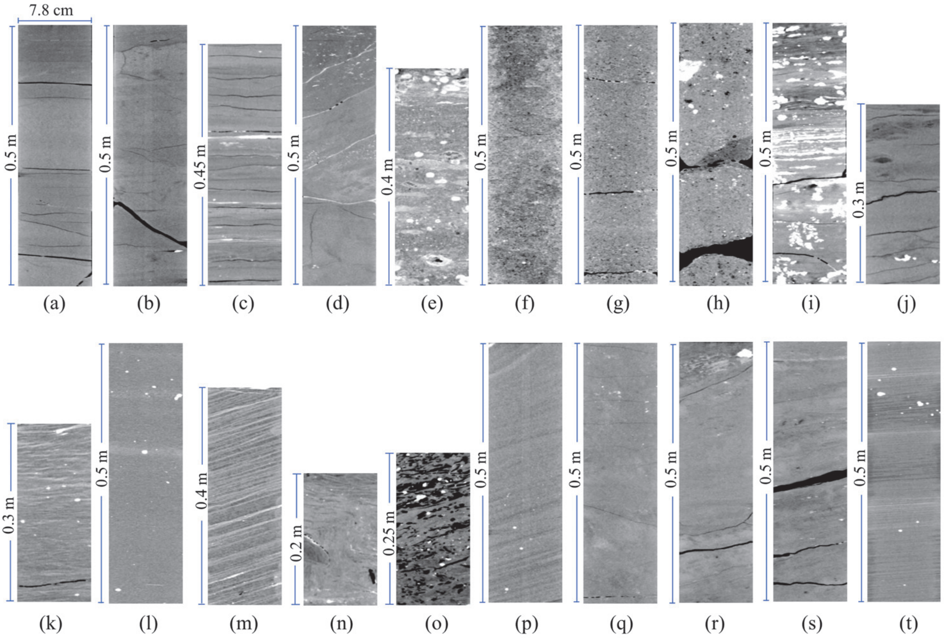


Figure 5.3: 2D cross-sectional images of lithofacies classes from manual core description: (a) Marl, (b) CalMarl, (c) SpiculiteSS, (d) Mudstone, (e) WCemBelSS, (f) GraMSSDispC, (g) PCemGraMSS, (h) WCemMSS, (i) MudsHighDens, (j) ArgFineSS, (k) RippleFineSS, (l) MassFineSS, (m) CrossFineSS, (n) MudFineSS, (o) BioFineSS, (p) WCemFineSS, (q) ContMud, (r) MassVeryFineSS, (s) CemVeryFineSS, (t) VeryFineSSHorizontal. A brief description of each lithofacies class in provided in Table 5.1.

mD up to 50 Darcy. Second well penetrates formation 3 and formation 4, where Formation 4 is characterized by lower porosity and permeability values with porosity ranging from 0.05 to 0.125, and permeability ranging from 0.01 to 1.7 mD. Formation 3 in the second well, covers a whole range of porosity values from 0.03 up to 0.32, and permeability values from 0.05 mD up to 50 darcy. Note that the lithology of Formation 3 in the second well is quite different than that of the first well, as it contains intervals with coarser grained lithology.

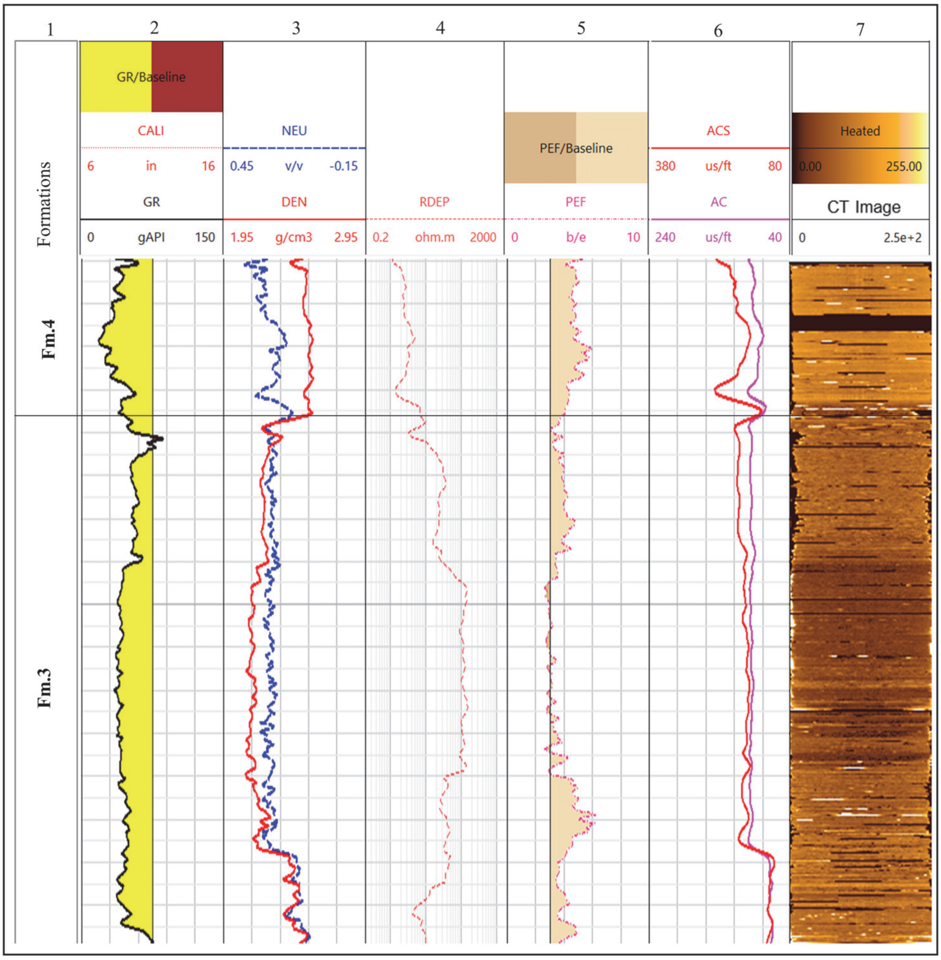


Figure 5.4: Well log data and 2D cross section of the core CT image showing 48.4 meters of the second well. Log tracks from left to right: track 1: Formations, track 2: Caliper (CALI) and Gamma ray (GR), track 3: Density (DEN) and Neutron (NEU), track 4: Deep resistivity (RDEP), track 5: Photoelectric factor (PEF), track 6: Compressional wave slowness (AC) and shear wave slowness (ACS), track 7: 2D cross section of whole core CT-scan.

Table 5.1: Lithofacies classes derived from core-based lithology descriptions.

Lithofacies labels	Description
Marl	Mud/clay rich marl
CalMarl	Marl with caliche cementation
SpiculiteSS	Medium-grained spiculitic sandstone
Mudstone	Dark gray mudstone with plain parallel bedding, mottled mudstone
WCemBelSS	Well cemented medium-grained sandstone with Belemnite fossils
GraMSSDispC	Granule rich medium-grained sandstone with dispersed carbonate cementation
PCemGraMSS	Poorly cemented granule rich medium-grained sandstone
WCemMSS	Well cemented medium-grained sandstone
MudsHighDens	Mudstone with high density minerals (pyrite)
ArgFineSS	Argillaceous fine-grained sandstone
RippleFineSS	Fine-grained sandstone with ripple cross lamination
MassFineSS	Massive fine-grained sandstone
CrossFineSS	Fine-grained sandstone with cross-stratified lamination
MudFineSS	Muddy fine-grained sandstone
BioFineSS	Bioturbated fine-grained sandstone
WCemFineSS	Well cemented fine-grained sandstone
ContMud	Continental mudstone
MassVeryFineSS	Massive very fine-grained green sandstone
CemVeryFineSS	Cemented very fine-grained green sandstone
VeryFineSSHorizontal	Very fine-grained sandstone with horizontal lamination

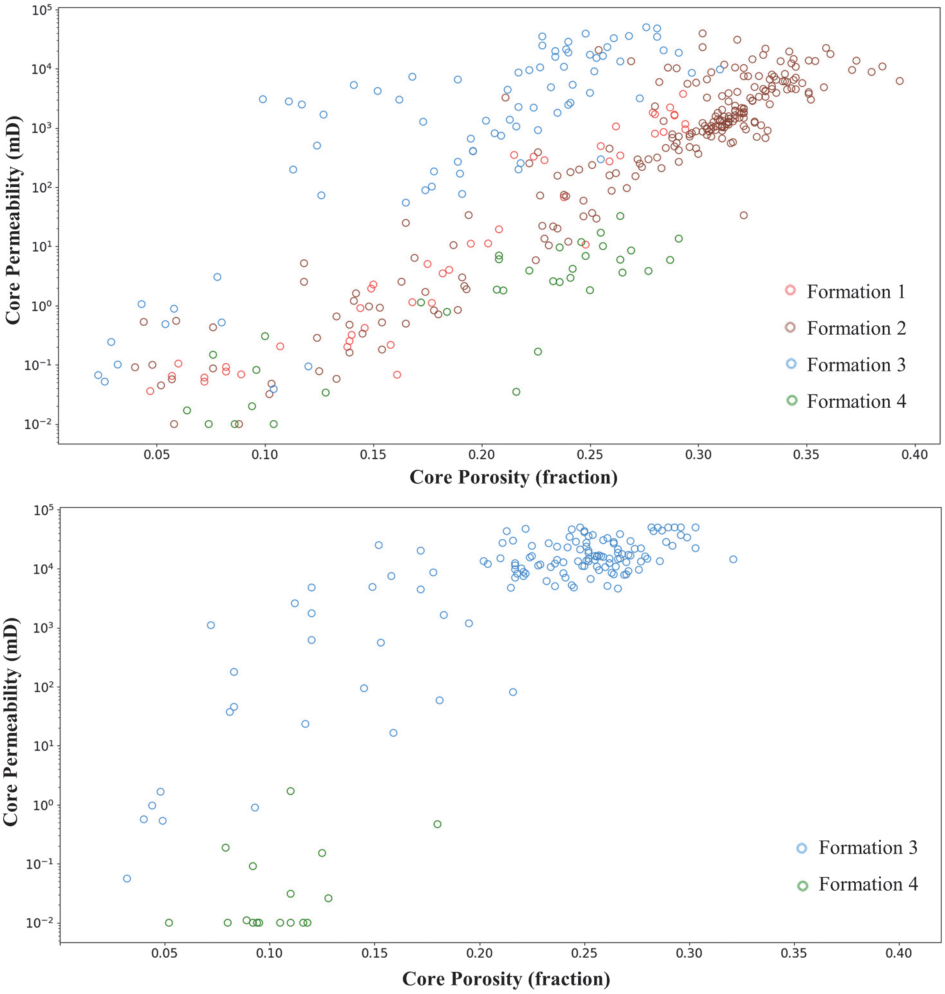


Figure 5.5: Porosity-permeability cross-plot for the first (above) and second (below) well. Different colors represent various formations penetrated by each well.

Chapter 6. Methods

This chapter presents details of image pre-processing, automatic depth shift of CT-scans, and the applied methodologies that we considered for solving the tasks of lithology classification and porosity estimation. The structure of the chapter reflects once again the logical structure of the considered tasks.

6.1 Image pre-processing

Images from certain intervals contain specific artefacts, features, and non-core regions that can negatively affect the performance of machine learning algorithms. Basically, these images contain information that is non-related to the actual phenomena we want to model. Therefore, they need to be flagged and removed before being used for further analysis. Examples of images with different types of artefacts are presented in Figure 6.1. In this regard, dedicated type-dependent processing steps were applied. These steps will be explained in the following sections.

6.1.1 Image cropping and intensity adjustment

The first adopted step is to remove border effects by cropping the cross-sectional and 2D image slices. A comparison of image examples before and after cropping is shown in Figure 6.2.

In case of 2D image slices, the original images are cropped into squared crops of size 256×256 pixels, as shown in Figure 6.2. Then, the cropped 2D image slices of individual cores are stacked together and stored as 3D raw images using the Imagej software (Schneider et al. 2012). Further, a global minimum and maximum intensity value, selected by observing the 3D histograms of all rectangular crops, is assigned to the images of the entire considered core intervals. The intensity adjusted images are, then, encoded in 8bit format, i.e., 0–255 gray-scale, and stored for further analysis.

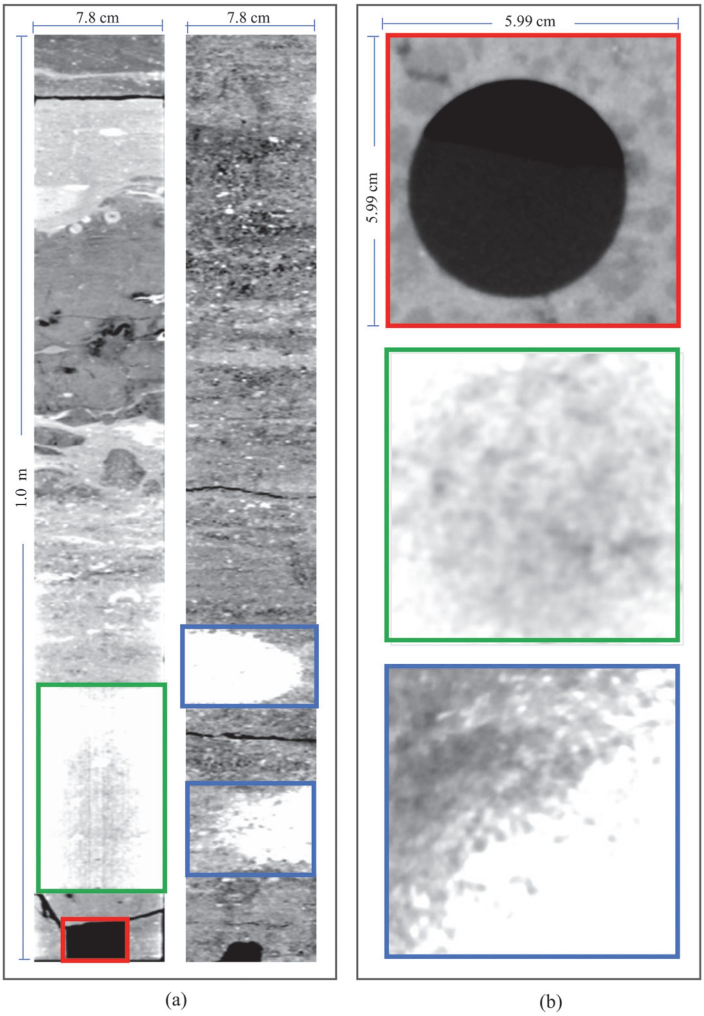


Figure 6.1: 2D cross-sectional (a) and image slices (b) with missing pixels (red rectangles), core barrel couplings (green rectangles), and high-density material (blue rectangles).

6.1.2 Flag and remove images with missing pixels

The considered dataset of CT images contains missing core intervals associated with poor core recovery, induced fractures, or rush plugs taken after the retrieval of the core. The missing pixels yield zero gray-level attenuation readings.

In case of cross-sectional images, the missing intervals are flagged using the fol-

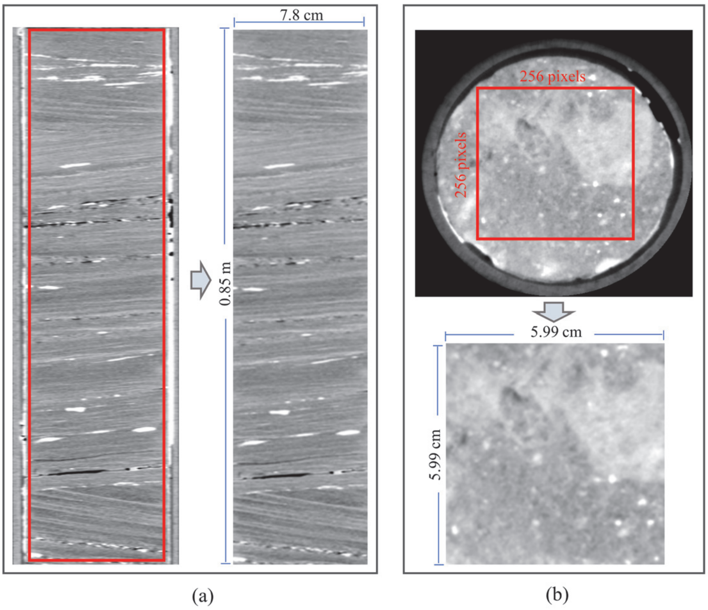


Figure 6.2: Image cropping in 2D cross-sectional (a) and image slices (b). The images are cropped to remove border effects and non-core regions.

lowing steps: We first identify the most frequently occurring gray-level attenuations, the mode M_o, at each depth, d. If the mode is 0 and the frequency of the mode f_{M_o} is more than 30% of the total number of pixels, N, then the image at that specific depth is flagged and removed from further analysis:

$$F_m(d) = \begin{cases} 1, & \text{if } M_o(d) = 0 \text{ and } f_{M_o}(d) > 0.3 \times N \\ 0, & \text{if } M_o(d) \neq 0 \text{ or } M_o(d) = 0 \text{ and } f_{M_o}(d) \leq 0.3 \times N \end{cases}, \tag{6.1}$$

where F_m is the missing flag in cross-sectional images. The image is removed when the missing flag, F_m, is equal to 1.

In case of 2D image slices, the images with missing data are flagged using the following strategy: We calculate the average attenuation μ_c in the center of the image using a centered square covering 40% of the total number of pixels. If the computed average attenuation is less than a pre-defined cutoff C_m, the image is flagged and removed:

$$F_{ms}(d) = \begin{cases} 1, & \text{if } \mu_c(d) < C_m \\ 0, & \text{else} \end{cases}, \tag{6.2}$$

where F_{ms} is the missing flag in image slices. The image is removed if F_{ms} is equal to 1.

6.1.3 Flag and remove images with core barrel couplings

In the intervals with core barrel couplings, the attenuation values in the middle of the images are lower than the attenuation values of the image edges (i.e., the edges are brighter, as in Figure 6.1).

To detect these artifacts in the cross-sectional images, at each depth, we compute the average attenuation at the sides of the image and compare it with the average attenuation in the middle. More specifically, at each depth we use the average of the outer 5% of the pixels to represent the outer "at-each-side" average μ_o, and the 20% of the total number of pixels in the middle of the image to represent the inner average attenuation μ_i of the image. If the difference between these average attenuations is greater than a predefined cutoff value C_b, the image at that specific depth is then flagged and removed:

$$F_b(d) = \begin{cases} 1, & \text{if } \mu_o(d) - \mu_i(d) > C_b \\ 0, & \text{else} \quad , \end{cases} \tag{6.3}$$

where F_b stands for core barrel coupling flag. Note that the cutoff C_b is defined by the user based on observing the attenuation differences between the center and sides of the images at depths with and without core barrel couplings. Note that the same strategy is used to detect 2D image slices with core barrel couplings. However, the inner average attenuation μ_i, in this case, is computed considering 40% of the total number of pixels using a centered square. Moreover, the outer 5% of the total number of pixels along the edges are considered to represent the outer average attenuation μ_o.

6.1.4 Flag and remove images with high density material

As shown in Figure 6.1, high density material such as mud invasion and high density cements appear as bright features with relatively high gray-level attenuation values. To detect images containing such features, the average attenuation μ of the whole image (either cross-sectional or image slice) is computed, and, if the average is greater than a pre-defined cutoff C_h, the image is flagged for removal:

$$F_h = \begin{cases} 1, & \text{if } \mu > C_h \\ 0, & \text{else} \end{cases} \tag{6.4}$$

where F_h is the high density flag. The image is removed if F_h is equal to 1.

Note that once the images with undesired features and artefacts are removed, the remaining images are coarsened by a factor of 4 to reduce the computational time associated with machine learning model training.

6.1.5 Preparing 3D images

In **paper III**, the 3D CT images were utilized to classify lithofacies. In this case, once the image slices with undesired features were removed from 3D core images of the individual cores, the remaining 3D images were coarsened by a factor of 4 along the width and height to reduce computational time needed for model training (i.e., the final image size along the width and height is 64×64 pixels). Note that no coarsening is performed along the length of the three-dimensional individual core images. Then, due to computational limitations associated with the amount of memory on the GPU card, sub-cubes of size $16 \times 16 \times 16$ pixels, with 50% overlap along all three dimensions, were extracted from the middle (covering 48 out of 64 pixels) of the processed individual whole core volumes. More precisely, for each depth we first extract a sub-cube of size $16 \times 16 \times 16$ pixels, then move the sub-cube by 8 pixels (50% overlap with the previous image), and then extract another sub-cube. This process is repeated until the whole image width (48 pixels) is covered. The same process is repeated along the height of the image. Once the entire cross section is covered, we move down in the depth direction by 8 pixels and extract new sub-cubes. Using this method, nine sub-cubes per depth interval are extracted and the lithofacies classes are assigned to the center pixel of the sub-cubes. This strategy results in a higher amount of training images, and it requires a smaller network that the available GPU memory is capable to handle.

6.2 Automatic depth shift of CT-scan images

Cores and wireline log data are two different types of measurements with their corresponding errors (e.g., differential cable stretch in logging tools) and every trip with the core barrel is potentially a change in the relative core or log depth, even if continuous cores are taken. This results in depth mismatches between cores and wireline logs that can destroy an otherwise good correlation between well logs and cores. Therefore, it is crucial to correct the depth differences when comparing the core-derived measurements and log responses.

In this study, we considered an automatic bulk depth shift of the cored sections using the bulk density log as the reference. For this purpose, we treat the problem as an optimization problem, where we minimize the difference between CT average attenuation and density log at each depth. The applied methodology can be summarized as follows:

1. Calculate average attenuation of 2D cross-sectional images at each depth:

   $$\bar{x}(d) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad , \tag{6.5}$$

   where $\bar{x}(d)$ is average attenuation at depth d, x_i is the i^{th} gray-level attenuation value, and N is the total number of pixels at depth d.

2. Scale the calculated average attenuation values: At this step, the acquired average attenuation values are scaled between initially guessed (by the user) minimum and maximum values.

3. Compute moving average of the scaled average attenuation values: As the vertical resolution of the density log ($\approx$ 30 cm) is lower than the vertical resolution of the whole core CT-scan images (≈ 0.45 mm), the scaled average attenuation values were coarsened to match the resolution of the density log. For this purpose, we calculated a moving average of the scaled average attenuation values using a window size of 30 cm corresponding to the resolution of the density log.

4. Interpolate bulk density log around the cored interval: Due to resolution differences (mentioned above), the frequency of data points in the density log is lower then the CT scan images, therefore, we interpolated the density log to construct intermediate density values. This will be required for computation of the cost function (next step).

5. Compute the cost function, Q, by:

   $$Q = \frac{1}{M}\sum_{i=1}^{M} |\bar{x}_{c_i} - \rho_{int_i}| \tag{6.6}$$

   where $\bar{x}_{c_i}$ is the coarsened average attenuation at depth i, ρ_{int_i} is the interpolated density at depth i, and M is total number of depth points.

6. At the last step, the above mentioned cost function is minimized, for which we employed the `Optimize` module from `SciPy` library (Virtanen et al. 2020) in Python. The `Optimize` module provides several commonly used optimization algorithms for unconstrained and constrained minimization of

> multivariate scalar functions. We specifically used the L-BFGS-B (Limited-memory Broyden–Fletcher–Goldfarb–Shanno Bound constrained) algorithm (Byrd et al. 1995; Zhu et al. 1997) to minimize the cost function. The L-BFGS-B algorithm is an extension of the L-BFGS (Liu and Nocedal 1989) algorithm that handles bound constraints on variables; that is, constraints of the form $l_i \leq x_i \leq u_i$ where l_i and u_i are per-variable constant lower and upper bounds for each x_i, respectively (Byrd et al. 1995). The method works iteratively by identifying the fixed and free variables at every step (using a simple gradient method), and then using the L-BFGS method only on the free variables to get higher accuracy. The lower and upper bound constraints are provided by the user. For more details on the L-BFGS-B algorithm, the interested reader is referred to (Byrd et al. 1995; Zhu et al. 1997).
>
> In our case, the lower and upper bound constraints for minimum scaled average attenuation, maximum scaled average attenuation, and vertical depth shift are provided by the user. The algorithm then minimizes the difference between the log density and CT values, and it returns the optimal values for above mentioned variables (i.e., minimum scaled average attenuation, maximum scaled average attenuation, and vertical depth shift).

This method was applied on the whole core CT-scans of the studied wells and the results are presented in the next chapter section 7.1.

6.3 Lithofacies classification

As mentioned, in this thesis three approaches were considered for lithology classification using 2D cross-sectional CT images, 2D images slices, and 3D images. The applied methodologies will be described in the following sections.

6.3.1 Lithofacies classification using feature extraction

The first proposed methodology (**paper I**) is summarized in Figure 6.3. This method utilizes the features extracted from 2D cross-sectional images as input to train an SVM whose goal is to classify lithofacies. Training data include information obtained from manual core descriptions (Table 5.1). More precisely, after image pre-processing, the remaining images were decomposed using Haar mother wavelet decomposition scheme (section 4.1) to obtain a set of detail images used to compute the first-order statistical features (section 4.2.1) and GLCM features (section 4.2.2). Here, three levels of wavelet decomposition were considered and the features were extracted from vertical detail images using a window size of 50 pixels (as described in Chapter 4, section 4.1, and section 4.2). Using this strategy, a total of 13 features were computed and used to represent a set of 2D cross-sectional images; four first-order statistical features (mean, variance, skewness, and kurtosis), and three GLCM features (contrast, energy, and correlation) for each of the three levels of decomposition. After feature extraction, the whole dataset was divided into train and test sets. The details of dataset division will be described in section 6.3.1.

After train-test split, we considered two workflows (Figure 6.3). In the first workflow, the extracted features of the train set were used as input to train an SVM to classify lithofacies. In addition, in the second workflow, we implemented an automated PCA step to perform an initial dimensionality reduction of the extracted features of the train set. This enables use of the principal components that capture most of the variation in the train set as the actual inputs to train the support vector classifier. The approach was then validated using the trained SVM and hybrid (PCA + SVM) models to predict the lithology in a set of unseen CT data (test set) and the results of both approaches were compared.

PCA implementation

As stated, a total of 13 features were extracted from decomposed vertical detail images. PCA was then used to perform dimensional reduction of the extracted features and to eliminate any possible correlation among them. The theory behind PCA is described in section 4.3.

Before performing PCA, the features were both mean-centered and normalized. Then a tenfold cross-validation approach was used to evaluate the PCA model. The correlation loading and score plots from this PCA step are shown in Figure 6.4. As stated before, in this study, three levels of decomposition were performed, and GLCM features were extracted from resulting vertical detail images. In Figure 6.4, L stands for the level of decomposition. By investigating the correlation loading plot, we can see the relative contribution of each feature in the computed principal components. According to this plot, the first principal component explains 36% of the variance in the feature space, whereas the second principal component captures 11% of the variation. We observe that GLCM features computed at three levels of decomposition are highly correlated. As an example, the energy features (i.e., Energy_L1, Energy_L2, and Energy_L3) are greatly correlated. Contrast and energy are negatively correlated, and together with variance they mostly contribute into the first principal component, whereas the second principal component is mainly

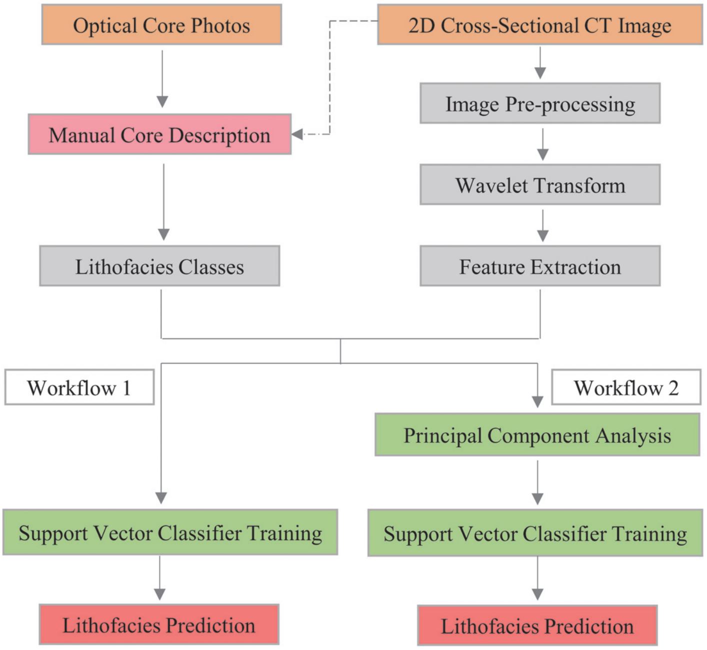


Figure 6.3: Proposed workflows for feature extraction and lithology classification in **paper I**. This illustration is a slightly modified version of Figure 1 in **paper I**.

affected by correlation, mean, skewness, and kurtosis.

Investigating the score plot, we can see that the considered lithofacies classes do not show a clear grouping or clustering even though some of the lithofacies classes with similar texture seem to lie closer to each other. Moreover, in the score plot, some of the argillaceous fine-grained sandstone samples (highlighted by red dashed ellipsoid in Figure 6.4) coming from image intervals with high density material appear as extreme observations characterized by high variance and contrast. It is worth to mention that we tried to perform the PCA analysis excluding these samples. However, the obtained PCA results did not change. The score plots of 20 individual lithofacies classes are presented in Figure 6.5 and Figure 6.6. Here, one can see that very fine-grained lithofacies classes, i.e., Marl, CalMarl, MassVeryFineSS, CemVeryFineSS, ContMud, and VeryFineSSHorizontal show low score values (between -5 and 5) along the first and second principal components. These classes are characterized by high energy, low variance and low contrast. Fine-grained lithofacies classes, e.g., RippleFineSS, CrossFineSS, and WCemFineSS show higher contrast and variance compared to the previously mentioned classes. Note that the majority of fine-grained samples share similarities with very fine-grained samples and they fall on top of each other for score values below 5 along both principal components. Some of the samples belonging to Mudstone, SpiculiteSS, and BioFineSS classes show higher skewness and kurtosis at the same time lower correlation and mean. Moreover, some of the samples of the Lithofacies classes containing high density material, i.e., MudsHighDens, GraMSSDispC, and ArgFineSS show higher score values along the first principal component due to their higher contrast, higher variance and lower energy. As mentioned before, despite differences among some of the samples from different lithofacies classes, most of the them fall on top of each other and they do not form distinctive clusters.

The explained variance curve (Figure 6.7) shows that approximately 93% of variance can be explained by nine principal components. Therefore, to use the reduced features, nine principal components were used as input to train a support vector classifier (paper I).

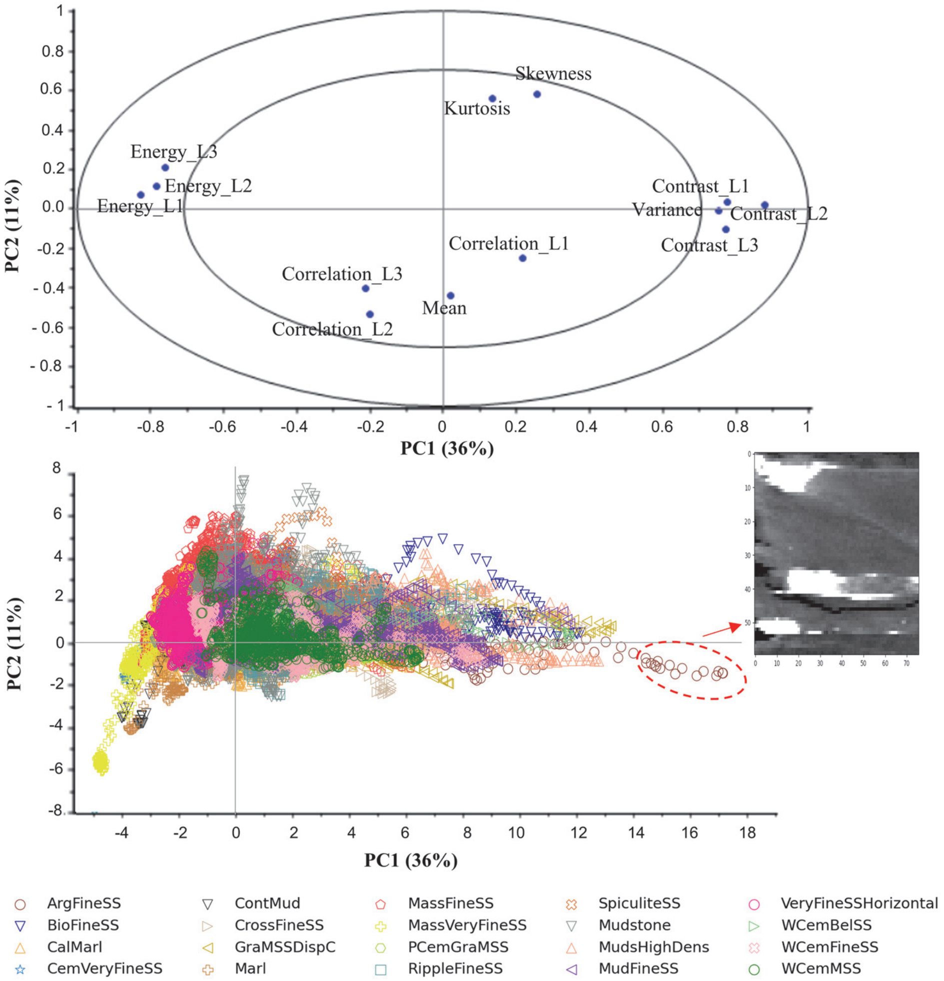


Figure 6.4: Correlation loading (top) and score (bottom) plots of the first and second principal components. Some of the ArgFineSS samples (highlighted by red dashed ellipsoid) from image intervals with high density material appear as extreme observations with high contrast.

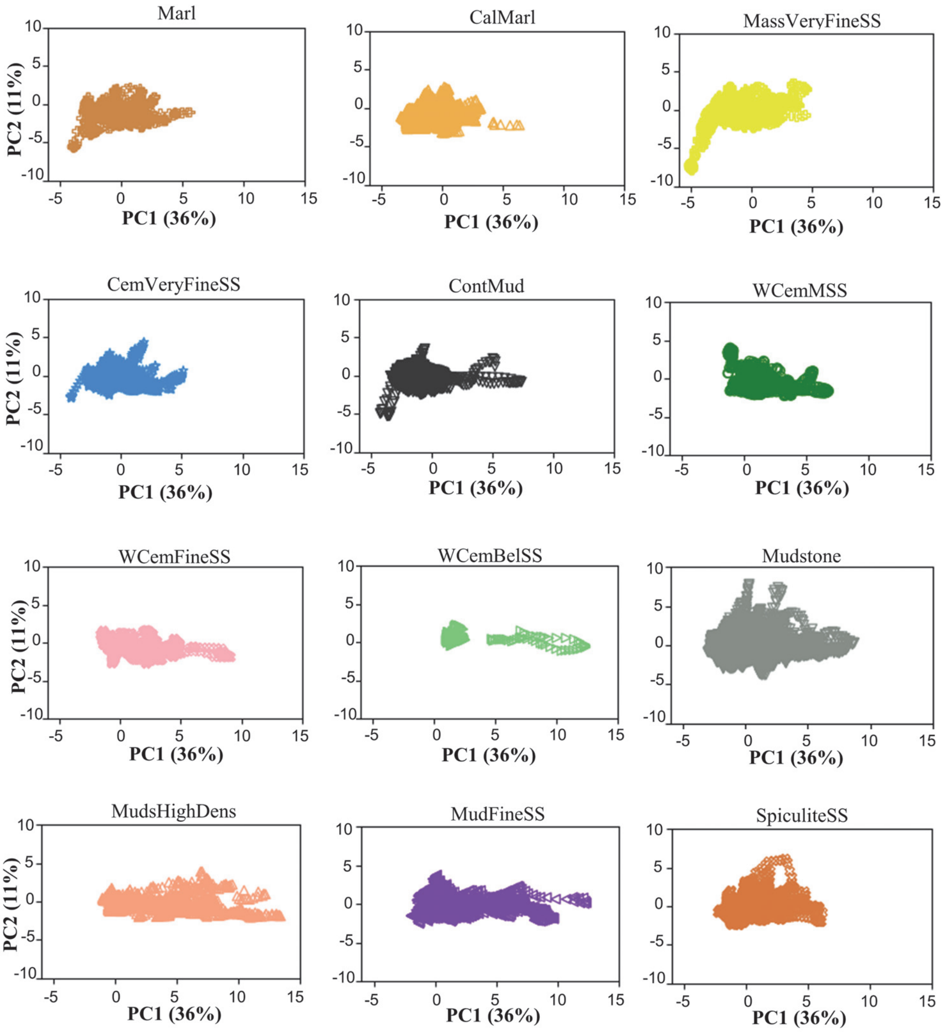


Figure 6.5: Score plots of the first and second principal components for individual lithofacies classes. Note that the score plots of other classes (not fitted in this figure) are presented in Figure 6.6.

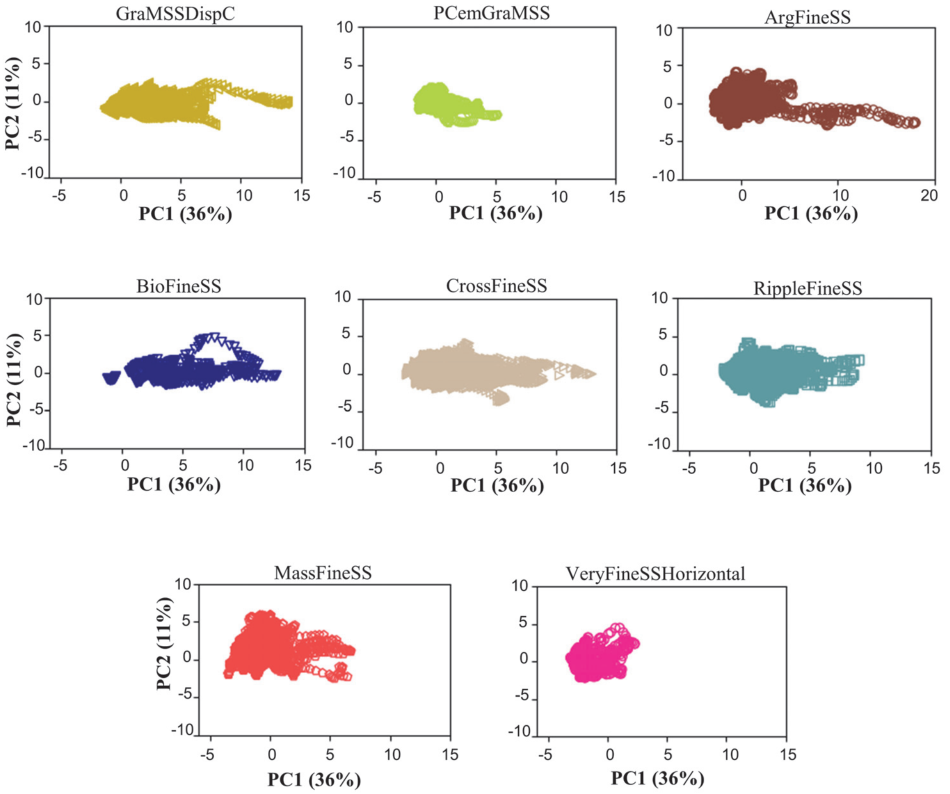


Figure 6.6: Score plots of the first and second principal components of other individual lithofacies classes (continued from Figure 6.5).

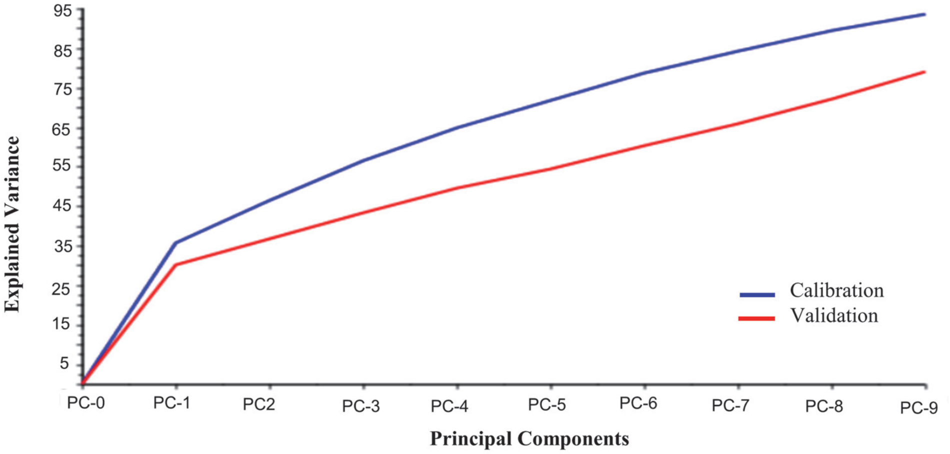


Figure 6.7: Explained variance plot from PCA analysis. This plot shows that approximately 93% of variance can be explained by nine principal components.

SVM training and hyperparameter selection

To prepare the dataset for SVM training and evaluation, the whole dataset was divided into train and test sets. Here, out of $60,415$ data points, we selected approximately 80% as train set and the remaining $\approx 20\%$ was used as the test set. The train and test sets were selected manually to maintain continuous intervals and at the same time keeping similar distribution of lithofacies classes in these two sets. The reason for not selecting train and test sets randomly is twofold. First, the GLCM features, in this study, were computed using a window size of 50 pixels; therefore, we need long intervals to maintain a large fraction of the data points; second, because the images are slowly varying, a random selection would give similar data points in the train and test sets. The extracted features were standardized before training, so that the first- and second-order (GLCM) statistical features have the same range of values.

As mentioned in section 4.4.1, SVM allows for misclassifications during the training phase by using a soft margin. This trades off bias and variance of the estimator through playing with a misclassification constant, c. This misclassification constant can be selected by performing a cross-validation to determine how many misclassifications are best to allow inside of the soft margin to get an expected best classification in future data. Another parameter that shall be tuned is the kernel that SVMs use to project the data into a higher dimensional feature space. Therefore, the choice of the kernel function can affect the results of the SVM classification to a very high extent. In this study, we used a Gaussian kernel (Eq. 4.29) and tuned the Gaussian kernel's width, γ, during the process of hyperparameter selection. The hyperparameters c and γ are usually selected through a trial-and-error process, in which a performance metric is used as the objective function. Here, these hyperparameters were optimized by means of a tenfold cross-validation process, in which the train set was partitioned into ten folds. In other words, different models each with a specific set of hyperparameters were trained on nine folds and evaluated on the remaining fold. The process was repeated ten times, whereby all the training instances got the chance to be used as both train and validation sets. The considered search space for c and γ values are presented in Table 6.1. The optimal values were selected based on the mentioned cross-validation scheme, and overall model average performance in predicting the cross-validated folds.

Table 6.1: Range of values tested for c and γ. The final optimal values are shown in bold. This table is taken from **paper I**.

Hyperparameter	Search space
c	$(10^{-3}, 10^{-2}, 10^{-1}, 1, 5, \mathbf{10}, 20, 50, 10^{2})$
γ	$(10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, \mathbf{1}, 10)$

6.3.2 Lithofacies classification using CNN

In the second (**paper II**) and third (**paper III**) approaches, we directly use the whole core CT images as input to the CNN models to learn the relationship between convolution-derived features and expert-derived lithofacies classes (Table 5.1). In these methodologies, the need for manual feature extraction is eliminated as relevant features are learned by the network while it is being trained on a set of CT images. The proposed methodologies are summarized in Figure 6.8 and Figure 6.9.

The methodology in **paper II** (Figure 6.8) starts with pre-processing of 2D DICOM images. Lithofacies labels are then assigned to the processed images based on expert-derived geological core description (section 5.2). The labeled images are divided into training, validation, and test sets. The images of the train set are further augmented and used as inputs to train a CNN classifier. The trained classifier is then validated on a set of unseen images to predict lithofacies classes. Once the prediction results are evaluated, a step of post classification processing is performed, where the lithofacies classes that are deemed to be sufficiently similar are combined together; in this step the similarity indexes are computed starting from assessments of the transport properties (porosity and permeability) together with the degree of confusion in the confusion matrix resulting from the learning algorithm. Further, the classifier is coarsened with respect to the number of rock classes; in other words, the classifier is trained with a smaller number of rock classes. Finally, the coarsened classifier is employed to predict rock classes on a set of unseen images.

As mentioned before, **paper III** utilizes the 3D CT-scan sub-cubes as input to train a CNN model to classify lithofacies. This paper concentrates on an interval of the well covering three lithofacies classes (i.e., GraMSSDispC, Mudstone, RippleFineSS). Here, once the images are pre-processed (i.e., 2D images with undesired features and artefacts are removed), the remaining images of each one-meter core interval are stacked together, and 3D sub-cubes of size 16 × 16 × 16 pixels are extracted as described in section 6.1.5. Lithofacies classes are assigned to the center pixel of each sub-cube, and they are divided into train and test sets. The sub-cubes of the train set are further augmented and used as input to train a CNN classifier to learn from manual core description. The trained model is then used to predict lithofacies classes on the images of the test set (Figure 6.9).

The details of image augmentation, CNN training, and hyperparameter selection process will be provided in the following sections.

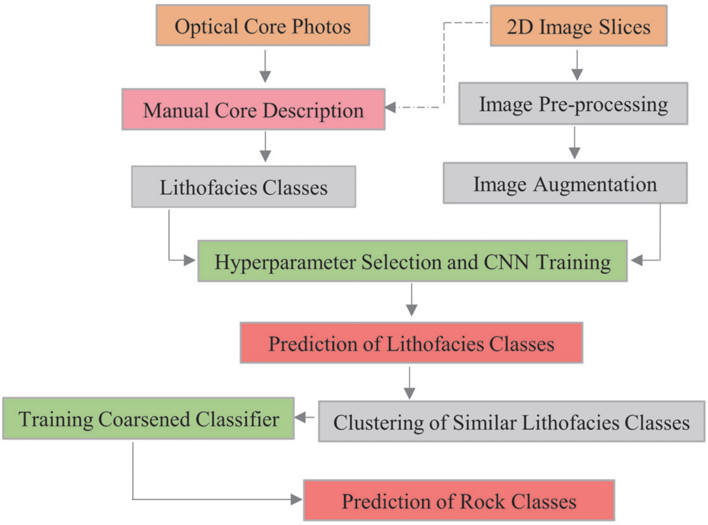


Figure 6.8: Proposed workflows for lithology classification using 2D image slices and CNN. This illustration is a slightly modified version of Figure 1 in **paper II**.

Image augmentation

Since large amounts of training data can lead to a better performance in deep neural networks, a step of image augmentation was considered in the approaches using the CNN algorithm (i.e., **paper II**, **paper III**, and **paper IV**). Image augmentation is a technique that is performed to boost the performance of the network through different kinds of modifications (e.g., random rotation, shifting, shearing and flipping) applied to the original images. Image augmentation is applied during the training phase, so that the model can learn from more image examples. In addition, invariance is incorporated into the model, and the risk of overfitting (section 4.4.2) is minimized. In this thesis, we specifically considered rotation and horizontal flip

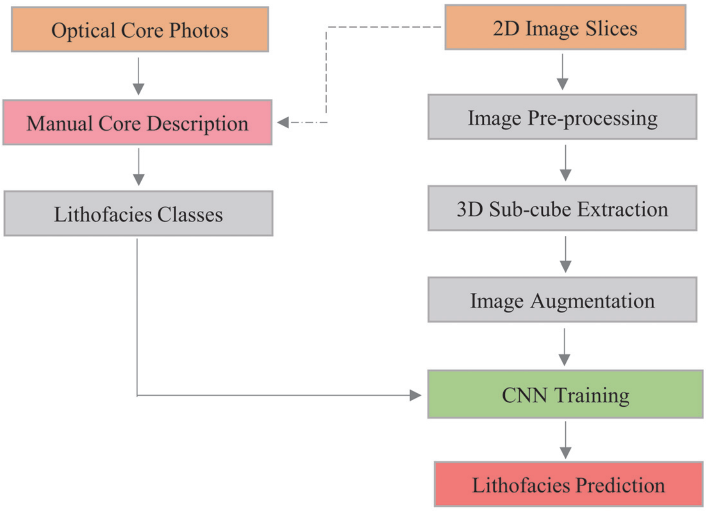


Figure 6.9: Proposed workflows for lithology classification using 3D images and CNN. This illustration is a slightly modified version of Figure 1 in **paper III**.

of the original images. For this purpose, we implemented the ”ImageDataGenerator” class in Python using the Keras API (Chollet et al. 2015), a publicly available code that can be used for image augmentation purposes on the fly. The ”ImageDataGenerator” class rotates the images randomly within a range of user-defined angles. Therefore, in case of squared images, it is very likely that for some specific rotation angles, the pixels will fall out of the image frame leaving some areas of the image with no pixels. There are a number of interpolation techniques such as nearest neighbour that can be used for those areas, but it can amend the key features resulting in dissimilar features counterproductive for training. To avoid this problem, the images were rotated outside Keras, while the horizontal flip was applied in Keras using ”ImageDataGenerator” class on the fly during training the CNN classifier. Here, the images were specifically rotated by $90^\circ, 180^\circ$, and 270°. An example of the rotated and horizontally flipped images is shown in Figure 4 in **paper II**. Note that in case of 3D images, the extracted 3D sub-cubes were rotated along the width and height, and no horizontal flip was considered in this case (**paper III**).

CNN training and hyperparameter selection

The CNN training occurs after splitting the dataset into train and test sets. Like the first paper, the dataset in the other two papers (i.e., papers on lithology classification) was divided into train and test sets using a splitting ratio of approximately 80% for training and 20% for testing. To maintain continuous intervals and at the same time keeping similar frequency (distribution) of the lithofacies between the sets, the train and test sets were selected manually. Moreover, approximately 20% of the train set was utilized as validation set, which was used to evaluate the performance of the CNN model during training.

The process of CNN training in general was described in Chapter 4, section 4.4.2. As stated previously, in addition to the learnable parameters, there are hyperparameters (e.g., learning rate, number of convolutional kernels, number of hidden layers and neurons in the fully connected layers) that are not learnable by network, and required to be set by the user. These hyperparameters can be adjusted during the process of hyperparameter tuning.

In this thesis, the CNN models were developed in Keras using the Tensorflow backend. Therefore, the hyperparameter tuning problem was solved using the Keras tuner library (O'Malley et al. 2019). This library enables to define a search space that includes the considered hyperparameters and a proper tuner that will automate the solution of the tuning process. More precisely, the task of the tuner is to evaluate a certain number of hyperparameter combinations in a model that is explicitly set-up for hypertuning, i.e., a *hypermodel*. Four tuners are available in Keras, including `RandomSearch`, `Hyperband`, `BayesianOptimization`, and `Sklearn`. For more information on the differences among these approaches we direct the interested reader to (Hutter et al. 2011; Snoek et al. 2012; Bergstra and Bengio 2012; Li et al. 2017). In this thesis we specifically utilized the Hyperband tuner (Li et al. 2017), which is a relatively new method for tuning the iterative algorithms. Basically, the strategy behind this approach is to try a large number of random configurations using adaptive resource allocation and an early stopping rule to quickly converge to a high performance model. More specifically, the random configurations are run for a specific number of epochs (i.e., one or two) per configuration, then the top-performing model configurations based on the previous results are trained for more epochs. Finally, the algorithm returns a best optimized configuration trained to the assigned maximum number of epochs. The best model configuration is chosen based on its performance on the validation set. The considered hyperparameters and optimized 2D and 3D classifier architectures, obtained by the mentioned hyperparameter selection process, are presented in the respective papers (i.e., Table 2 and Figure 7 in **paper II**; Table 2 and Figure 4 in **paper III**). Note that the cross-entropy loss (Eq. 4.33) and a batch size consisting of 32 image samples were used in these two papers.

It is worth to mention that the employed CNN architecture in **paper III** utilizes the global average pooling (GAP) technique (Figure 4 in **paper III**), where the GAP layer replaces the fully connected layers. This architecture was chosen to reduce the risk of overfitting, number of free parameters and subsequent computational time for training and hyperparameter tuning.

6.4 Porosity estimation

In regards to porosity estimation (**paper IV**), we utilized 2D image slices of scanned whole cores to predict porosity using the RCA-derived porosity measurements as the ground truth. The applied methodology is summarized in Figure 6.10. Here, we compared two approaches for automatic porosity estimation at the core scale. Like the previous methods, the methodology here starts with pre-processing of 2D image slices, where images with undesired features and artefacts are automatically flagged and removed.

After image pre-processing the remaining images are divided into the train and test sets. Then, two approaches are considered: In the first, the images of the train set are augmented and together with porosity values from routine core analysis are employed to train a CNN regression model. In the second approach, a simple linear regression (LR) model is trained to learn the relationship between average attenuations of 2D images and RCA-derived porosity values. The reason for choosing a linear model is that we expect a linear correlation between the average gray-level attenuations and porosity for a monomineralic rock type with one type of fluid. This comparison reveals if the CNN model is capable of learning the distribution of gray-level attenuations (in form of extracted features) rather than just learning the average attenuation of the images. The training results of the two approaches are compared, and finally the best performing model, i.e., the CNN, is used to predict porosity of unseen images and to populate the whole well with millimeter scale porosity values.

6.4.1 Labeling procedure and train-test splitting

As stated, the RCA-derived core plug porosity measurements were used as the target porosity values for individual 2D images. Porosity is often measured on core plugs typically 1 to 1.5 in. (2.5 to 3.8 cm) in diameter and 1 to 2 in. (2.5 to 5 cm) long. The length of core plugs correspond to approximately 55 to 110 2D image slices with vertical resolution of 0.45 millimeters. In our approach we assign the same porosity labels to 19 successive 2D images (approximately 8.5 mm vertical length) at depth intervals corresponding to the core plug depths. In other

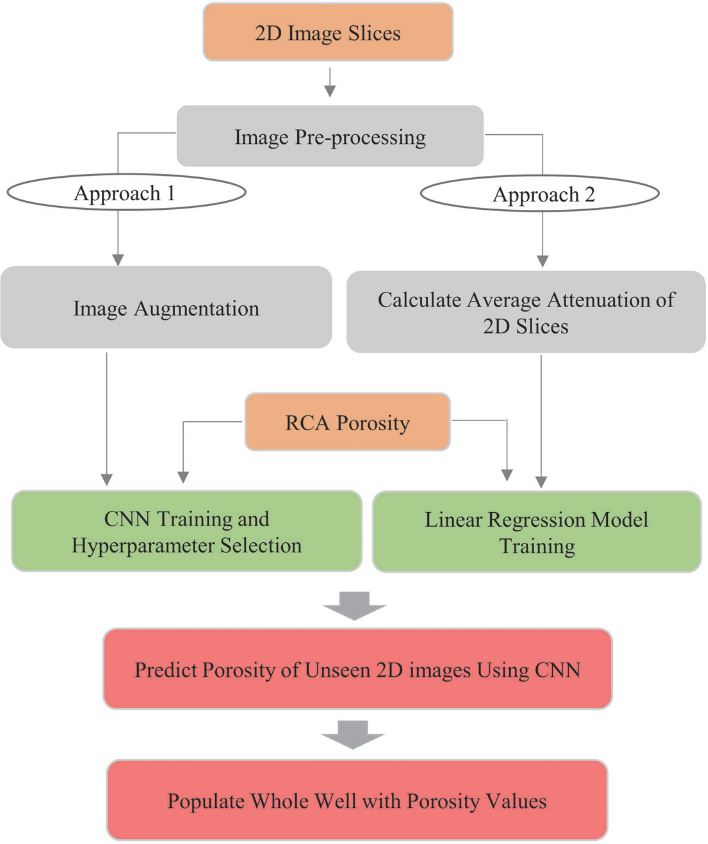


Figure 6.10: Proposed workflows for porosity estimation. This illustration is a slightly modified version of Figure 1 in **paper IV**.

words, individual image slices are used as input to train the CNN model, where 19 successive images at each depth interval are labeled with the same porosity values coming from core plug measurements. Note that the number of images in specific intervals might be less than 19 since images with artefacts are removed by image pre-processing.

For linear regression modeling we use the same labeling strategy. However, in this case the average attenuation of each 2D image is computed, and the model is trained to learn the relationship between average attenuations (as independent variables) and core analysis porosity measurements (as dependent variables).

The aforementioned labeling strategy resulted in 6863 labeled images. To cover the distribution of the measured porosity values, we chose $\approx 90\%$ of the images as the training set and the remaining $\approx 10\%$ as the test set. Moreover, 20% of the training set was employed as the validation set to provide an unbiased evaluation of the model while tuning. Further augmentation (image rotation by three angles of 90°, 180°, and 270°) of the training images resulted in 19764 training images. Note that the image augmentation is only applied on the images of the training set. The rotated images of the training set are also horizontally flipped on the fly and during training.

6.4.2 CNN regression training and hyperparameter selection

Similar to **paper II** and **paper III**, the CNN model in **paper IV** was developed in Keras, and the hyperparameters were tuned using the Keras Tuner library (section 6.3.2). The details of considered hyperparameters and the final optimized network architecture are provided in **paper IV**.

Note that here we are dealing with a regression problem, where the CNN model is trained to predict continuous porosity values. As mentioned before, MSE (Eq. 4.34) and MAE (Eq. 4.35) are typically considered as the cost function in problems dealing with prediction of continuous values. In this thesis we considered both types of the above cost functions. However, based on our preliminary experiments the optimization of MAE led to a model with better performance on the validation set than a model optimized with MSE. Therefore, we used MAE as the cost function to be minimized during model training in **paper IV**.

6.4.3 Linear regression training

As indicated, in a second approach we trained a linear regression model to predict RCA porosity from the average attenuation values calculated for individual 2D image slices. The purpose of such a step was to enable a subsequent analysis revealing if the CNN model extracts features rather than just learning the average attenuation of the images. The calculated average attenuations versus measured porosity values for the training dataset are shown in Figure 6.11. Note that the same porosity values are assigned to 19 successive image slices. We can see that the number of images associated to some porosity values are less than 19 since some of the images with artefacts are removed by image pre-processing. This figure shows that porosity is negatively correlated with average attenuation. However, some of the data points (see red and blue ellipsoids in Figure 6.11) behave differently and appear as outliers. Our investigations show that the samples within the red ellipsoid are associated with images that partly contain missing pixels. The data points within the blue ellipsoid, instead, are associated with images containing high density material. Apparently, images containing these features (i.e., missing pixels and high density materials) were not completely removed by the image pre-processing step. The images with missing pixels are characterized by lower gray-level attenuation values, resulting in lower average attenuation compared to the other images with similar measured porosity values, whereas images with high density material show higher average attenuation values compared to the images with similar measured porosity values.

Since the linear regression training process can be affected by outliers, a step of outlier detection and removal was included prior to model training. To detect the outliers, we employed the Isolation Forest (iForest) algorithm initially proposed by Liu et al. (2008, 2012). The iForest algorithm is a model that is based on unsupervised learning, and works using the principle of isolating anomalies. Similar to the Random Forest, the iForest method randomly splits the data points by building an ensemble of trees (called iTrees), where the goal here is to isolate the anomalous data points. Based on this algorithm, there is a tendency for anomalous instances to be isolated easier compared to the normal instances, i.e., anomalies are the data points with short average path lengths on the iTrees. The outlier detection results will be presented in section 7.4.1.

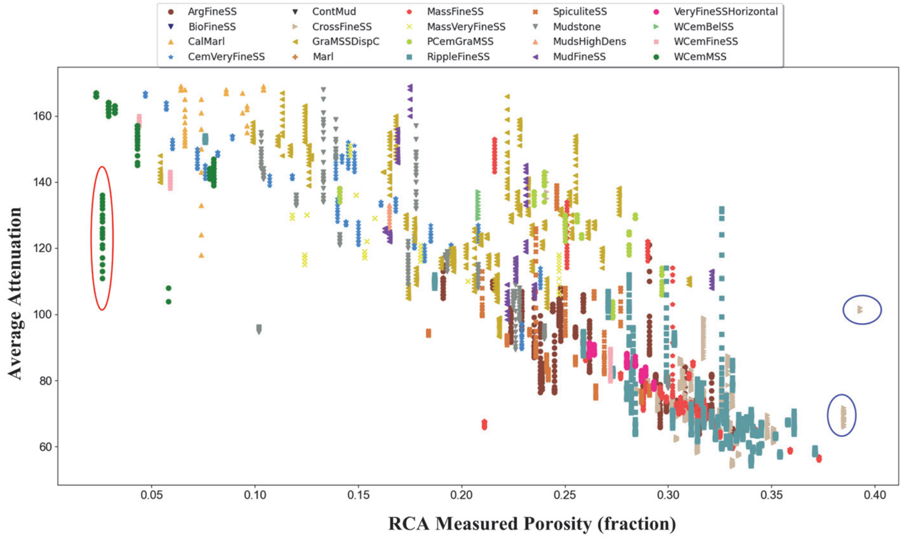


Figure 6.11: Average attenuation of 2D image slices versus RCA-derived porosity measurements. Different colors correspond to different lithofacies in the first well. The data points within the red and blue ellipsoids are considered among the possible outliers. This figure is taken from **paper IV**.

Chapter 7. Results and discussions

This chapter summarizes and connects the findings from the scientific papers (i.e., **papers I** to **IV**). The chapter is divided into six main sections. The first section provides the acquired results on the automatic depth shift of the core CT-scan images of the studied wells. The results of lithology classification using 2D (i.e., **paper I** and **paper II**) and 3D images (**paper III**) are discussed in the second and third sections, respectively. Section four then deals with results on porosity estimation (i.e., **paper IV**). The employed CNN architecture and the performance of the proposed architecture in **paper II** will be compared with a deeper architecture (i.e., VGG16) in the fifth section. The last section of this chapter discusses the train-validation split strategy and its influence on the model performance.

7.1 Automatic depth shift results

As mentioned in section 6.2, we developed a method for automatic depth shift of the whole core CT-scan images. The methodology was applied on the studied wells and the results are presented in Figure 7.1 and Figure 7.2, where we can see reasonable shifts of the CT-scan images with respect to the density log as the reference log. The acquired results are satisfactory given that we are applying a bulk shift on the entire cored section. Note that, due to applied bulk shift, there might be local depth mismatches as individual cores might need different depth shifts.

7.2 Results on lithology classification using 2D images

As mentioned, **paper I** and **paper II** propose workflows for lithology classification using 2D cross-sectional and 2D image slices, respectively. More specifically, in the first paper two approaches were considered to classify lithology based on extracted statistical features, i.e., one only based on an SVM, and the other applying a PCA before the SVM step, referred to as the hybrid model. In the second

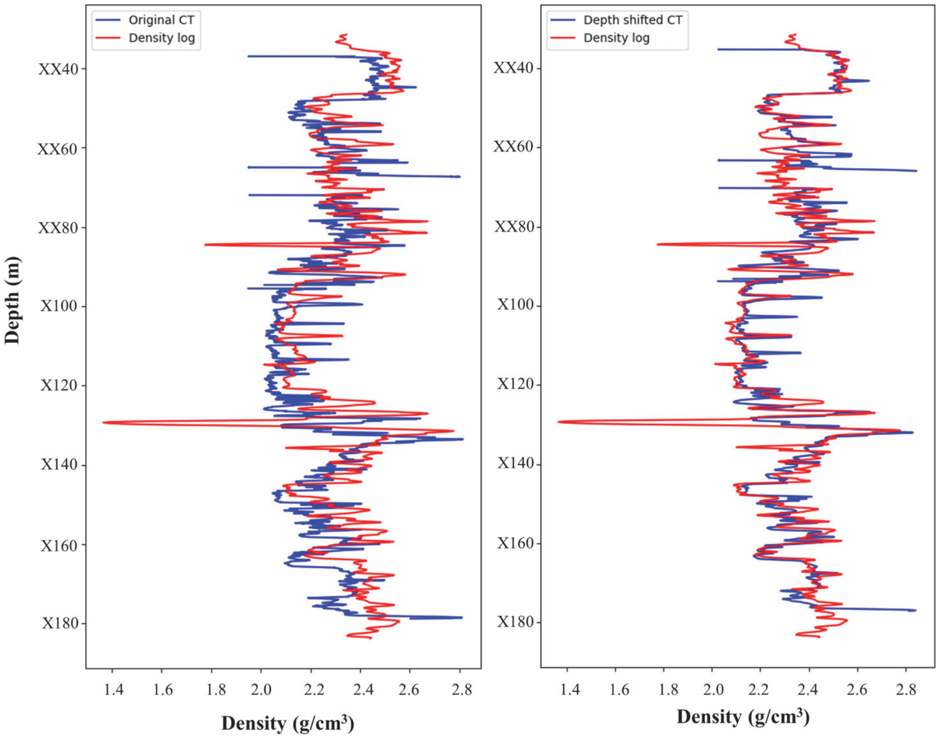


Figure 7.1: Original (left) and depth shifted (right) CT-scan images of the first well plotted with the density log as the reference log. The optimal depth shift in this well is -1.73 meters.

paper the 2D image slices were directly used as input to train a CNN classifier to learn lithofacies from expert-derived manual core description.

To evaluate the prediction performance of the proposed methods (in both papers), the trained classifiers were used to predict lithofacies on other sections of the first well completely unseen by the models. The test dataset went through the same processes of image pre-processing before being used for lithology prediction.

The accuracy metrics of the SVM and hybrid classifiers are compared in Table 7.1. Note that in case of the hybrid classifier, nine principal components, resulted from PCA analysis (section 6.3.1), were selected as input features to train an SVM classifier. The extracted statistical features in the test set were projected onto this new coordinate system (i.e., PCA model), and the trained hybrid model was used to predict lithofacies in the test set.

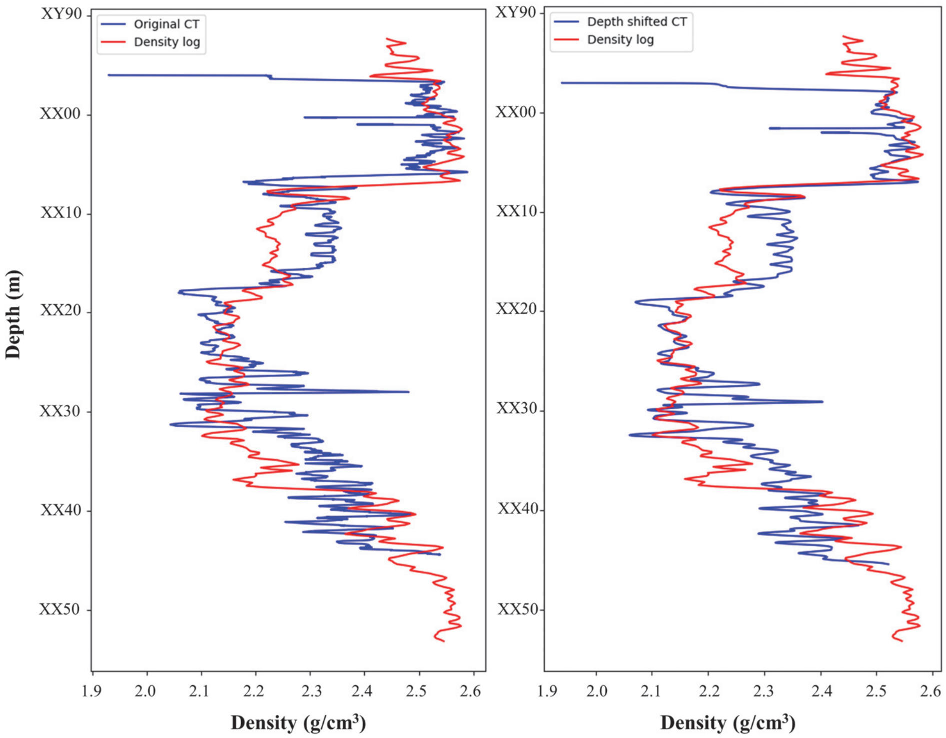


Figure 7.2: Original (left) and depth shifted (right) CT-scan images of the second well plotted with the density log as the reference log. The optimal depth shift in this well is +1.009 meters.

Here, precision is quantified as the sum of true positives divided by the sum of true positives and false positives across all the lithofacies classes in the test set. In other words, precision represents the probability that the predicted lithofacies class, given the classification results for individual images, actually belongs to that class. Recall is calculated as the sum of the true positives divided by the sum of true positives and false negatives across all the lithofacies classes. Precision and recall results are combined into a single measurement, i.e., the f1-score, through the following formula:

$$\text{f1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \; . \tag{7.1}$$

The obtained results, (Table 7.1), clearly show that the SVM model outperforms

	Precision		Recall		F1-score	
Lithofacies Labels	SVM	Hybrid	SVM	Hybrid	SVM	Hybrid
Marl	0.46	0.03	0.45	0.03	0.45	0.03
CalMarl	0.62	0.00	0.84	0.00	0.71	0.00
SpiculiteSS	0.35	0.01	0.53	0.01	0.43	0.01
Mudstone	0.46	0.16	0.63	0.26	0.53	0.20
WCemBelSS	0.32	0.00	0.27	0.00	0.29	0.00
GraMSSDispC	0.80	0.06	0.91	0.03	0.85	0.04
PCemGraMSS	0.79	0.28	0.70	0.31	0.74	0.29
WCemMSS	0.61	0.06	0.47	0.03	0.53	0.04
MudsHighDens	0.54	0.06	0.49	0.04	0.52	0.05
ArgFineSS	0.34	0.01	0.42	0.03	0.38	0.01
RippleFineSS	0.27	0.02	0.36	0.01	0.31	0.01
MassFineSS	0.83	0.03	0.76	0.03	0.79	0.03
CrossFineSS	0.64	0.20	0.49	0.18	0.55	0.19
MudFineSS	0.22	0.12	0.11	0.04	0.15	0.06
BioFineSS	0.45	0.02	0.30	0.01	0.36	0.01
WCemFineSS	0.21	0.00	0.06	0.00	0.09	0.00
ContMud	0.49	0.16	0.46	0.05	0.48	0.07
MassVeryFineSS	0.48	0.02	0.33	0.01	0.39	0.01
CemVeryFineSS	0.36	0.07	0.20	0.04	0.26	0.05
VeryFineSSHorizontal	0.83	0.01	0.45	0.01	0.58	0.01

Table 7.1: Comparison of prediction metrics of the SVM and hybrid (PCA+SVM) classifiers on the test set. The SVM clearly outperforms the hybrid classifier.

the hybrid model. This indicates that the generalization capabilities of the classifier diminish by using fewer principal components. Using nine principal components may not be enough to explain the variation in the dataset. The score plots and correlation loading plots of the remaining low-power principal components (i.e., PC10, PC11, PC12, and PC13) are plotted versus the first principal component in Figure 7.3, 7.4, 7.5, and 7.6. Overall, these four last principal components explain 7% of variance of the feature variables. The correlation loading plot in Figure 7.3 shows that most of the variables, except contrast and variance, do not show significant contribution to the PC10. The score plot shows that some observations belonging to the ContMud, WCemBelSS, CrossFineSS, and MudsHighDens classes are characterized by higher variance, while specific samples from ArgFineSS class are responsible for relatively higher contrast and lower variance. The most influencing variables for PC11, PC12, and PC13 are contrast and energy. Note that even though these lower power principal components do not capture much of the variance in the dataset, the amount of variance captured by them might still be important to distinguish between different lithofacies. In other words, all information in the CT images is required to differentiate lithofacies classes. Henceforward, the prediction performance of the SVM classifier (**paper I**) will be compared to the performance of the CNN classifier (**paper II**).

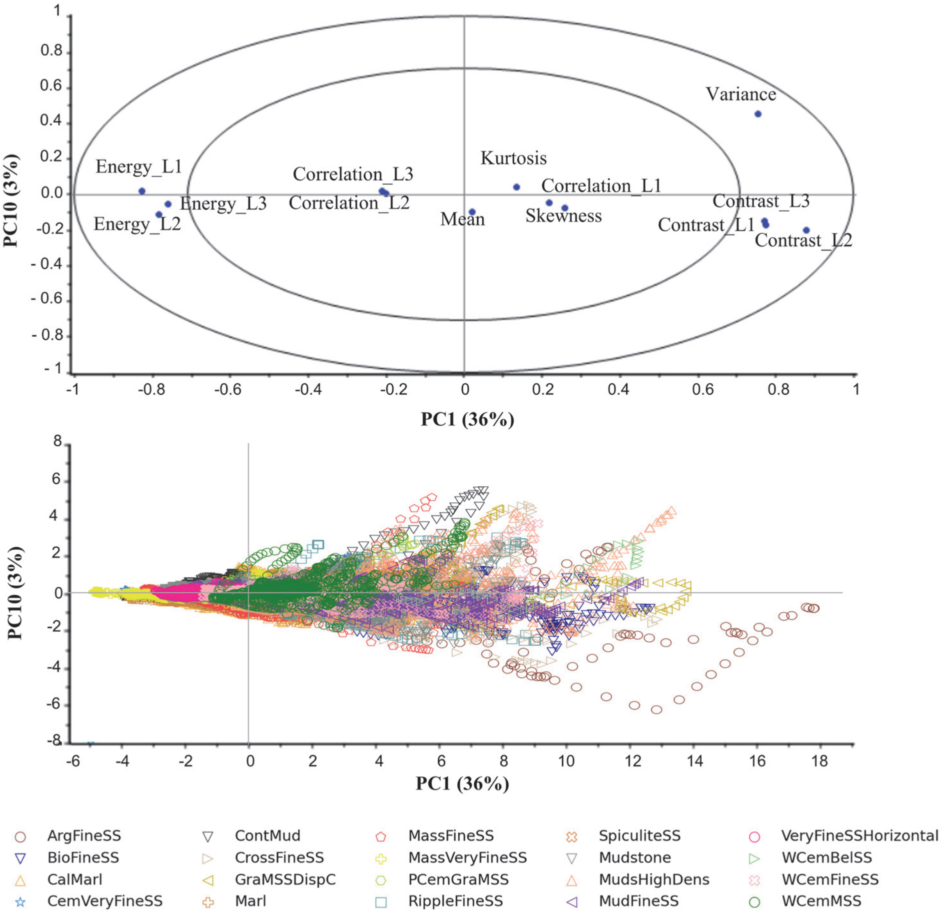


Figure 7.3: Correlation loading and score plots of PC1 versus PC10.

The prediction confusion matrix of the SVM classifier is presented in Figure 7.7. This matrix is calculated by cross-classifying the actual lithofacies (i.e., lithofacies from manual core description) and the predicted lithofacies classes. Recall is shown in the diagonal of the confusion matrix, whereas the off-diagonal values correspond to the degree of confusion (or probability of misclassifications). Accord-

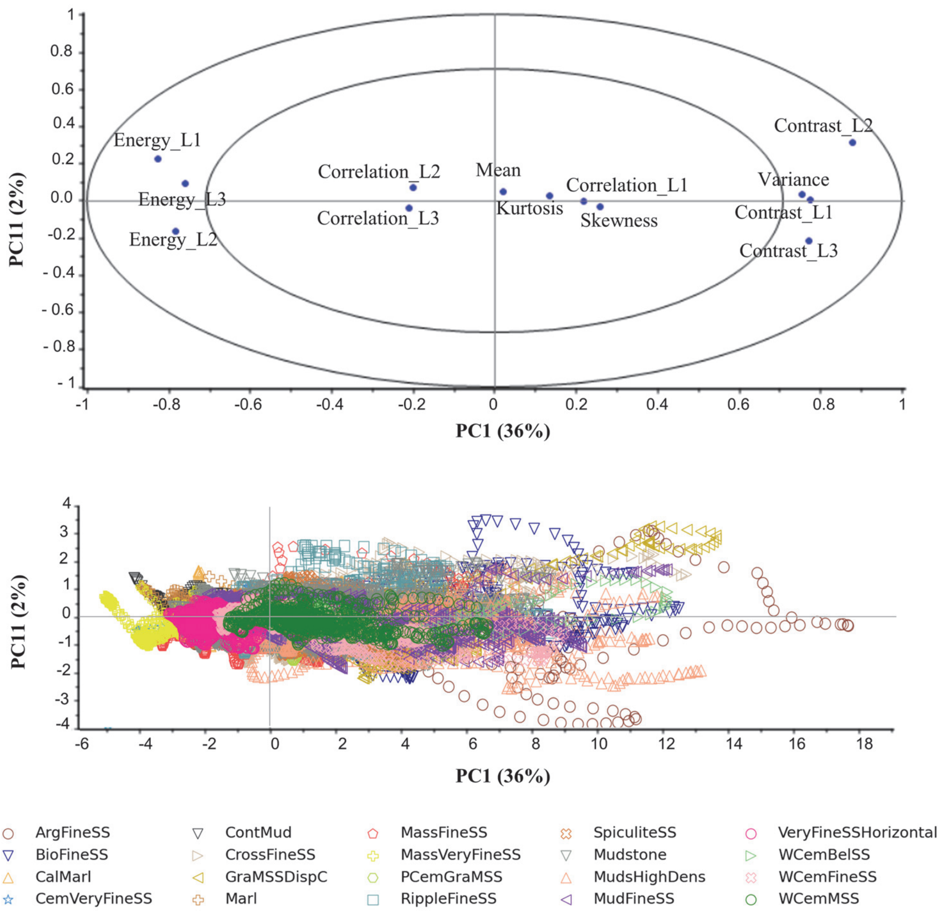


Figure 7.4: Correlation loading and score plots of PC1 versus PC11.

ing to Figure 7.7, the SVM classifier predicts some of the lithofacies classes with a recall of more than 0.7 (e.g, CalMarl, GraMSSDispC, PCemGraMSS, and MassFineSS), whereas the majority of the lithofacies classes are misclassified. Overall, the SVM classifier shows an accuracy of 0.55.

The prediction confusion matrix of the CNN classifier (**paper II**) is shown in Figure 7.8. Like the SVM classifier, we observe that the CNN classifier predicts some of the lithofacies with higher recall. However, it classifies other lithofacies into another class or a set of classes with various degrees of confusion. The overall accuracy of the CNN classifier is 0.56, slightly better than the SVM classifier (i.e.,

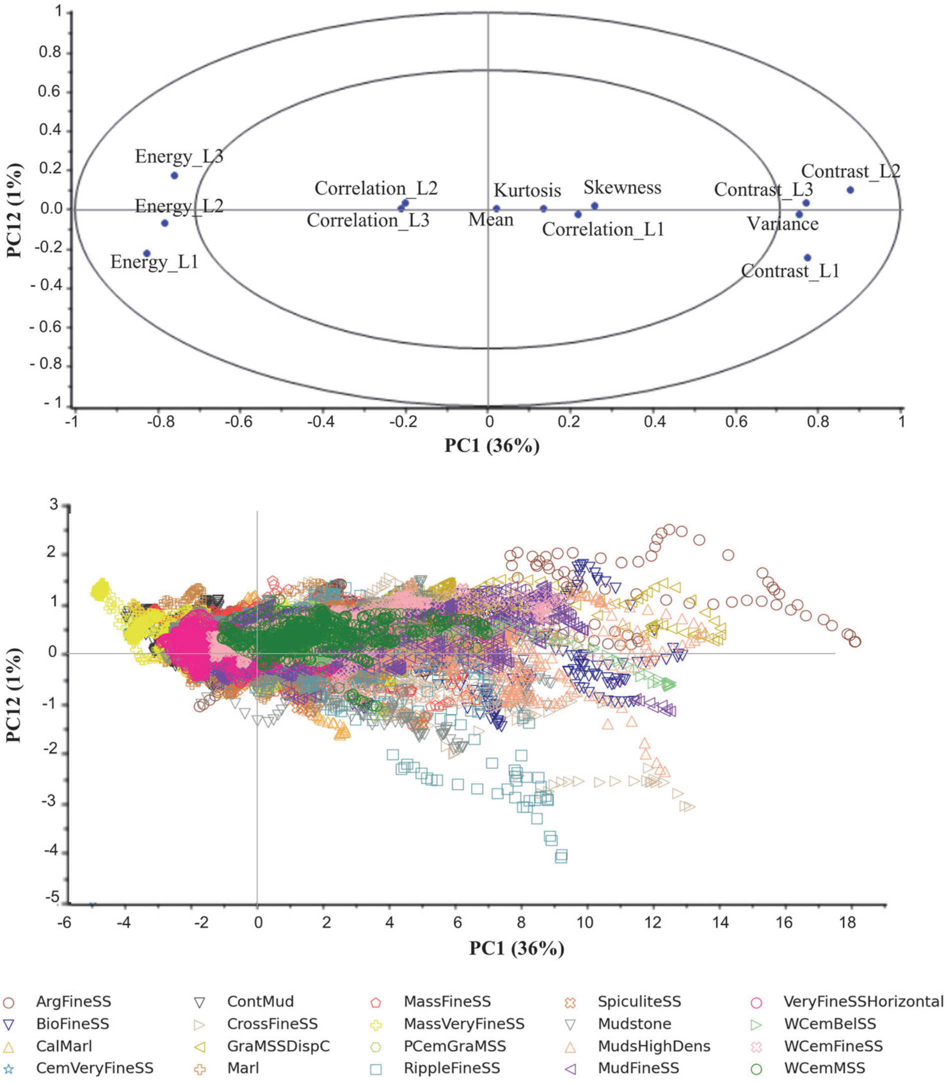


Figure 7.5: Correlation loading and score plots of PC1 versus PC12.

with overall accuracy of 0.55).

Examples of misclassified images are shown in Figure 9 and Figure 10 in **paper I** and **paper II**, respectively. These images mainly belong to very fine-grained lithofacies classes (i.e., marl, marl with caliche cementation, mudstone, mudstone with

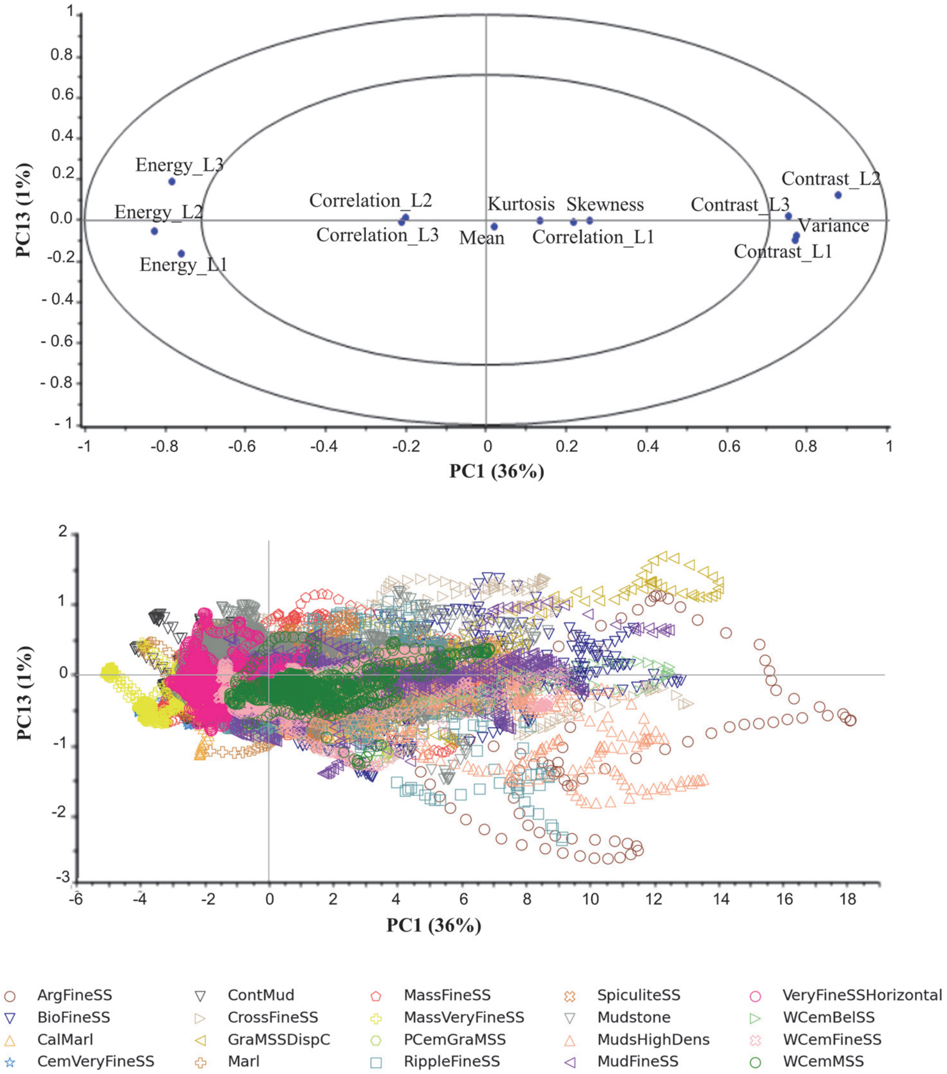
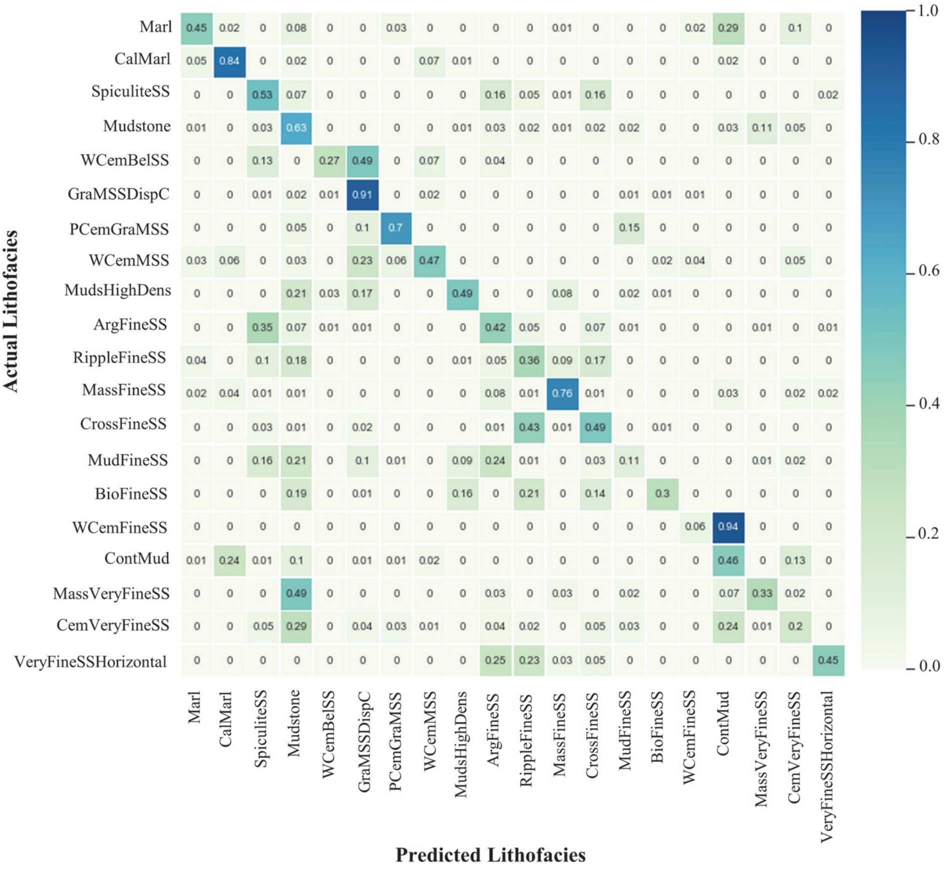


Figure 7.6: Correlation loading and score plots of PC1 versus PC13.

high density minerals, muddy fine-grained sandstone, cemented very fine-grained green sandstone, massive very fine-grained green sandstone, and continental mudstone) with similar texture and grain sizes, therefore similar grayscale values, with no distinct features. Likewise, our investigations showed that fine-grained sandstones with horizontal lamination, ripple cross lamination and cross-stratified lam-

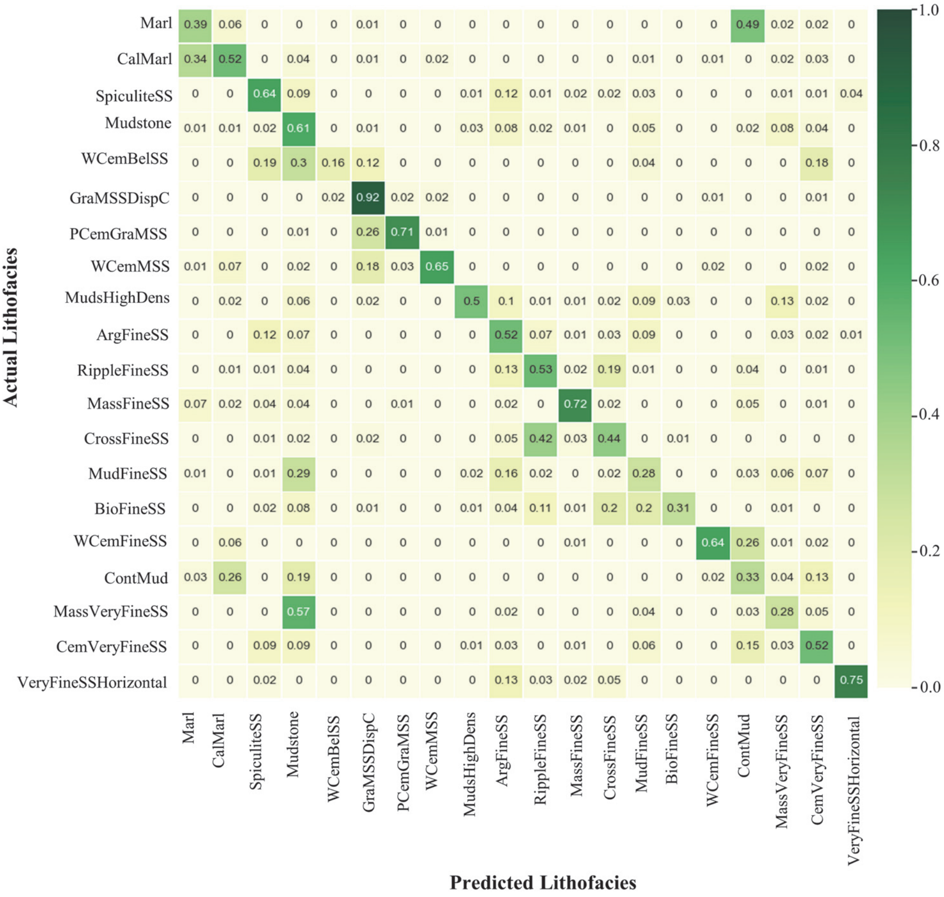


	Marl	CalMarl	SpiculiteSS	Mudstone	WCemBelSS	GraMSSDispC	PCemGraMSS	WCemMSS	MudsHighDens	ArgFineSS	RippleFineSS	MassFineSS	CrossFineSS	MudFineSS	BioFineSS	WCemFineSS	ContMud	MassVeryFineSS	CemVeryFineSS	VeryFineSSHorizontal
Marl	0.45	0.02	0	0.08	0	0	0.03	0	0	0	0	0.01	0	0	0	0.02	0.29	0	0.1	0
CalMarl	0.05	0.84	0	0.02	0	0	0	0.07	0.01	0	0	0	0	0	0	0	0.02	0	0	0
SpiculiteSS	0	0	0.53	0.07	0	0	0	0	0	0.16	0.05	0.01	0.16	0	0	0	0	0	0	0.02
Mudstone	0.01	0	0.03	0.63	0	0	0	0	0.01	0.03	0.02	0.01	0.02	0.02	0	0	0.03	0.11	0.05	0
WCemBelSS	0	0	0.13	0	0.27	0.49	0	0.07	0	0.04	0	0	0	0	0	0	0	0	0	0
GraMSSDispC	0	0	0.01	0.02	0.01	0.91	0	0.02	0	0	0	0	0	0.01	0.01	0.01	0	0	0	0
PCemGraMSS	0	0	0	0.05	0	0.1	0.7	0	0	0	0	0	0	0.15	0	0	0	0	0	0
WCemMSS	0.03	0.06	0	0.03	0	0.23	0.06	0.47	0	0	0	0	0	0	0.02	0.04	0	0	0.05	0
MudsHighDens	0	0	0	0.21	0.03	0.17	0	0	0.49	0	0	0.08	0	0.02	0.01	0	0	0	0	0
ArgFineSS	0	0	0.35	0.07	0.01	0.01	0	0	0	0.42	0.05	0	0.07	0.01	0	0	0	0.01	0	0.01
RippleFineSS	0.04	0	0.1	0.18	0	0	0	0	0.01	0.05	0.36	0.09	0.17	0	0	0	0	0	0	0
MassFineSS	0.02	0.04	0.01	0.01	0	0	0	0	0	0.08	0.01	0.76	0.01	0	0	0	0.03	0	0.02	0.02
CrossFineSS	0	0	0.03	0.01	0	0.02	0	0	0	0.01	0.43	0.01	0.49	0	0.01	0	0	0	0	0
MudFineSS	0	0	0.16	0.21	0	0.1	0.01	0	0.09	0.24	0.01	0	0.03	0.11	0	0	0	0.01	0.02	0
BioFineSS	0	0	0	0.19	0	0.01	0	0	0.16	0	0.21	0	0.14	0	0.3	0	0	0	0	0
WCemFineSS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0.94	0	0	0
ContMud	0.01	0.24	0.01	0.1	0	0.01	0.01	0.02	0	0	0	0	0	0	0	0	0.46	0	0.13	0
MassVeryFineSS	0	0	0	0.49	0	0	0	0	0	0.03	0	0.03	0	0.02	0	0	0.07	0.33	0.02	0
CemVeryFineSS	0	0	0.05	0.29	0	0.04	0.03	0.01	0	0.04	0.02	0	0.05	0.03	0	0	0.24	0.01	0.2	0
VeryFineSSHorizontal	0	0	0	0	0	0	0	0	0	0.25	0.23	0.03	0.05	0	0	0	0	0	0	0.45

Figure 7.7: Confusion matrix for the test set prediction using the SVM classifier. This figure is from **paper I**.

ination were also confused by the above classifiers (i.e., SVM and CNN) due to similarities in the grain size, texture, and grayscale values.

As lithofacies with similar grayscale and textural properties are expected to exhibit similar transport properties, porosity and permeability data from core analysis measurements were used to investigate the transport properties of different lithofacies classes. Figure 7.9 shows the porosity-permeability cross-plot of the core plug samples derived from the same core data as in the CT images. In this figure different colors correspond to different lithofacies classes that have been derived from the manual core description. Our investigations revealed that misclassified



	Marl	CalMarl	SpiculiteSS	Mudstone	WCemBelSS	GraMSSDispC	PCemGraMSS	WCemMSS	MudsHighDens	ArgFineSS	RippleFineSS	MassFineSS	CrossFineSS	MudFineSS	BioFineSS	WCemFineSS	ContMud	MassVeryFineSS	CemVeryFineSS	VeryFineSSHorizontal
Marl	0.39	0.06	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0.49	0.02	0.02	0
CalMarl	0.34	0.52	0	0.04	0	0.01	0	0.02	0	0	0	0	0	0.01	0	0.01	0	0.02	0.03	0
SpiculiteSS	0	0	0.64	0.09	0	0	0	0	0.01	0.12	0.01	0.02	0.02	0.03	0	0	0	0.01	0.01	0.04
Mudstone	0.01	0.01	0.02	0.61	0	0.01	0	0	0.03	0.08	0.02	0.01	0	0.05	0	0	0.02	0.08	0.04	0
WCemBelSS	0	0	0.19	0.3	0.16	0.12	0	0	0	0	0	0	0	0.04	0	0	0	0	0.18	0
GraMSSDispC	0	0	0	0	0.02	0.92	0.02	0.02	0	0	0	0	0	0	0	0.01	0	0	0.01	0
PCemGraMSS	0	0	0	0.01	0	0.26	0.71	0.01	0	0	0	0	0	0	0	0	0	0	0	0
WCemMSS	0.01	0.07	0	0.02	0	0.18	0.03	0.65	0	0	0	0	0	0	0	0.02	0	0	0.02	0
MudsHighDens	0	0.02	0	0.06	0	0.02	0	0	0.5	0.1	0.01	0.01	0.02	0.09	0.03	0	0	0.13	0.02	0
ArgFineSS	0	0	0.12	0.07	0	0	0	0	0	0.52	0.07	0.01	0.03	0.09	0	0	0	0.03	0.02	0.01
RippleFineSS	0	0.01	0.01	0.04	0	0	0	0	0	0.13	0.53	0.02	0.19	0.01	0	0	0.04	0	0.01	0
MassFineSS	0.07	0.02	0.04	0.04	0	0	0.01	0	0	0.02	0	0.72	0.02	0	0	0	0.05	0	0.01	0
CrossFineSS	0	0	0.01	0.02	0	0.02	0	0	0	0.05	0.42	0.03	0.44	0	0.01	0	0	0	0	0
MudFineSS	0.01	0	0.01	0.29	0	0	0	0	0.02	0.16	0.02	0	0.02	0.28	0	0	0.03	0.06	0.07	0
BioFineSS	0	0	0.02	0.08	0	0.01	0	0	0.01	0.04	0.11	0.01	0.2	0.2	0.31	0	0	0.01	0	0
WCemFineSS	0	0.06	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.64	0.26	0.01	0.02	0
ContMud	0.03	0.26	0	0.19	0	0	0	0	0	0	0	0	0	0	0	0.02	0.33	0.04	0.13	0
MassVeryFineSS	0	0	0	0.57	0	0	0	0	0	0.02	0	0	0	0.04	0	0	0.03	0.28	0.05	0
CemVeryFineSS	0	0	0.09	0.09	0	0	0	0	0.01	0.03	0	0.01	0	0.06	0	0	0.15	0.03	0.52	0
VeryFineSSHorizontal	0	0	0.02	0	0	0	0	0	0	0.13	0.03	0.02	0.05	0	0	0	0	0	0	0.75

Figure 7.8: Confusion matrix for the test set prediction using the CNN classifier trained on 2D image slices. This figure is from **paper II**.

lithofacies classes fall into the same region showing similar porosity-permeability relationships.

Based on similarities in transport properties, the 20 lithofacies classes can be grouped into 4 clusters marked by ellipsoids in Figure 7.9. In this figure, the aforementioned very fine-grained lithofacies classes fall into the red ellipsoid characterized by porosity and permeability values less than 0.20 and 10 mD, respectively. Likewise, fine-grained sandstones, with different types of laminations (i.e., horizontal, ripple cross lamination, and cross-stratified), together with massive fine-

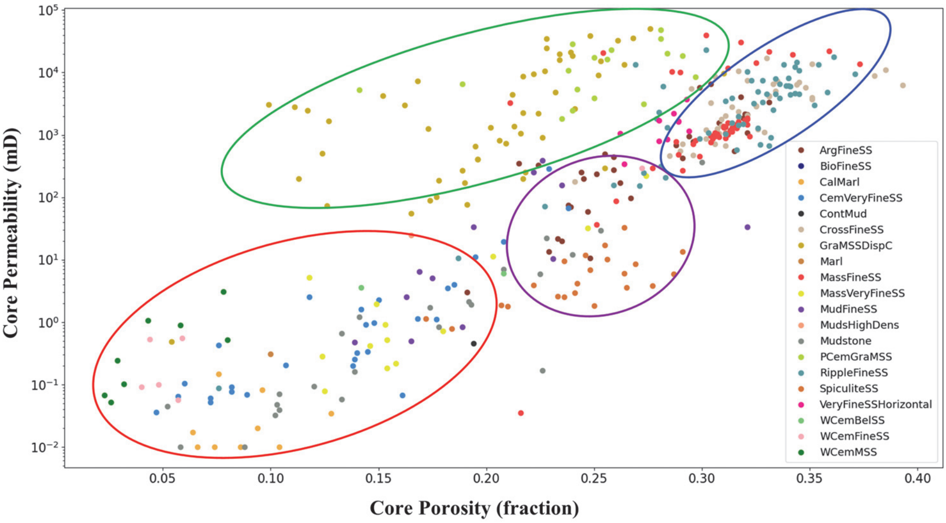


Figure 7.9: Porosity-permeability cross-plot from available core measurements for the first well. The misclassified lithofacies show similar porosity permeability trends marked by ellipsoids with different colors. This figure is from **paper II**.

grained sandstones show similar porosity-permeability relationships, represented by blue ellipsoid. More precisely, these lithofacies classes exhibit porosity values above 0.28 and permeability values ranging from 100 mD to approximately 30 Darcy. Granule-rich medium-grained sandstone samples (i.e., PCemGraMSS, GraMSSDispC) spread out over a wide range of permeability values from 30 mD up to 50 Darcy, marked by a green ellipsoid. However, the majority of samples belonging to these classes characterize by porosity and permeability values above 0.20 and 1 Darcy, respectively. The CNN prediction results (Figure 7.8) indicate that poorly cemented granule-rich sandstone (PCemGraMSS) lithofacies are mainly misclassifed as granule-rich sandstone with dispersed calcite cementation (GraMSSDispC).

Moreover, the spiculite sandstone samples exhibit porosity values ranging from 0.20 to 0.28 and permeability values from 1 mD to 20 mD. The CNN classifier mostly misclassifies the spiculite class as argillaceous fine-grained sandstone. These two classes show similar porosity-permeability trends and most of the samples belonging to these classes fall within the purple ellipsoid. We note that the SVM classifier mostly misclassifies the spiculite sandstones as cross-stratified fine-grained sandstones and argillaceous fine-grained sandstones with similar degrees of confusion. The porosity-permeability cross-plot shows that some of the measurements belonging to the cross-stratified and argillaceous fine-grained lithofacies classes spread out in relatively larger range of porosity and permeability values and they fall within both the blue and purple ellipsoids. This shows that some of the images belonging to the cross-stratified sandstone lithofacies show similar transport properties as spiculite sandstones, and might explain the reason for model confusion.

For more quantitative investigations of the acquired prediction results (in **paper I** and **paper II**), we fit a log-linear regression line to map the porosity-permeability relationships. The purpose here was to calculate the intercept values using a fixed regression coefficient. The differences in the intercept values can then be used as an indication of differences among various lithofacies. The computed intercept values are presented in Figure 11 and Figure 12 in **paper I** and **paper II**, respectively. Note that the intercept values in **paper I** are calculated using a fixed regression coefficient for each cluster (blue and red ellipsoids in Figure 10 in the first paper), whereas the intercept values in **paper II** are computed per lithofacies classes and for classes with more than five measurements. This is the reason for different intercept values for the same lithofacies classes in these two papers. In both papers, we can clearly infer that most of the lithofacies with similar transport properties tend to group into similar sets of intercept values.

Given the similarities in transport properties of the misclassified lithofacies, it is not unreasonable to expect classification confusion amongst these classes.

It is worth mentioning that generalization capability of the SVM model, from the first paper, was further evaluated by employing the model to predict lithofacies in an interval of the second well. The interested reader is referred to **paper II** for more details on the obtained results.

7.2.1 Post-classification processing

The acquired results, from the first two papers, revealed that the prediction confusion matrix can provide valuable information about the similarities and relationships between different lithofacies classes. Moreover, these results indicate that the confused lithofacies classes can be grouped into four clusters based on their similarities in porosity-permeability relationships.

In **paper II**, the information provided by prediction confusion matrix and porosity-permeability trends was used as a tool to post-process the acquired CNN prediction results. More precisely, we considered coarsening of the classification task with respect to the number of lithofacies classes as a post-classification processing step. For this purpose, inspired by Godbole (2002), we used the information acquired from the confusion matrix to automatically generate lithofacies hierarchies based on the degree of confusion for the different lithofacies classes.

Hierarchical clustering is a method in clustering analysis that aims at building a hierarchy of clusters based on a predefined similarity metric. The basic theory and different types of hierarchical clustering are described in **paper II**, section 6.1. In this thesis, we performed hierarchical clustering using the empirical confusion matrix from the CNN classifier as the quantitative measure of distance between the various lithofacies. This corresponds to use an Euclidean distance as the inter-class similarity metric between lithofacies class vectors in the confusion space. More precisely, the Euclidean distance is, in our work, calculated by summing up the absolute differences in the coordinate values of two class vectors. To exemplify the process, consider the confusion matrix in Table 7.2, where for simplicity we show only the results relative to four classes. Each class is represented by a vector in the confusion space, i.e., $\overrightarrow{GraMSSDispC} = \{0.92, 0.02, 0, 0\}$ represents the GraMSSDispC lithofacies class. The Euclidean distances mentioned above are then calculated by summing up the absolute differences in the coordinate values of the class pairs. In this way it is possible to compute an upper triangular similarity matrix as the one shown in Table 7.3. This, in particular, shows that RippleFineSS and CrossFineSS classes are the most similar ones among the set of classes considered in this sub-confusion matrix used to exemplify the process.

Table 7.2: Confusion matrix of four classes. Here we consider four classes for simplicity. This table is taken from **paper II**.

	GraMSSDispC	PCemGraMSS	RippleFineSS	CrossFineSS
GraMSSDispC	0.92	0.02	0	0
PCemGraMSS	0.26	0.71	0	0
RippleFineSS	0	0	0.53	0.19
CrossFineSS	0.02	0	0.42	0.44

The computations in the example above are then performed in the original complete confusion matrix, and the resulting similarity matrix is used as the input for the hierarchical agglomerative clustering step.

Table 7.3: Similarity matrix computed using the confusion matrix in Table 7.2. This table is taken from **paper II**.

	GraMSSDispC	PCemGraMSS	RippleFineSS	CrossFineSS
GraMSSDispC	0	1.35	1.66	1.78
PCemGraMSS	-	0	1.69	1.81
RippleFineSS	-	-	0	0.38
CrossFineSS	-	-	-	0

The output of this clustering step is a dendrogram shown in Figure 7.10. This dendrogram shows the overall results of clustering similar lithofacies classes based on their degree of confusion in the CNN confusion matrix. Note that the vertical axis in the dendrogram is used as a reference distance that shows the similarity of the lithofacies classes. This means that the plot shows not only how different the classes are, but also the order by which lithofacies clustering occurs. We note that the obtained dendrogram, for most of the classes, clearly reflects the semantic similarity of the lithofacies in the confusion space confirmed by various degrees of confusion in Figure 7.8.

However, as an example, we notice that the well cemented medium-grained sandstone (WCemMSS) class is first merged with the poorly cemented granule rich sandstone class (PCemGraMSS), and at a slightly higher level they merge with granule rich sandstone with dispersed cementation (GraMSSDispC). Recall that it was previously shown that the granule rich lithofacies core measurements spread out in the regions with high permeability values ranging from 30 mD up to 50 Darcy (i.e., the green ellipsoid in Figure 7.9), where the majority of samples exhibit porosity and permeability values above 0.20 and 1 Darcy, respectively. On the other hand, the well cemented sandstone samples (WCemMSS) are characterized by porosity and permeability values less than 0.10 and 5 mD, respectively. Merging these classes, with completely different transport properties, does not seem reasonable.

Based on the hierarchical clustering results and porosity-permeability relationships, the original classification task was coarsened with respect to the number of classes. More precisely, the original 20 lithofacies classes were merged into four rock classes, as presented in Table 7.4. Then, the ground truth lithofacies la-

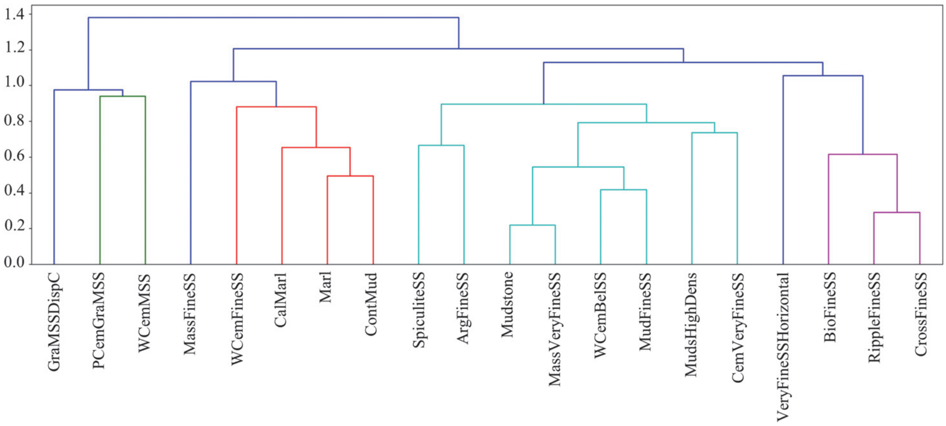


Figure 7.10: The resulting dendrogram from agglomerative hierarchical clustering step. The dendrogram represents the clustering of lithofacies classes together using a distance measure calculated from the CNN confusion matrix. This figure is taken from **paper II**.

bels, derived from manual core description, were modified to reflect these 4 rock classes instead of original 20 classes. This implies that one can re-train the original CNN classifier proposed above using this new set of labels, and also perform a new round of testing.

The resulted prediction confusion matrix of the test set is shown in Figure 7.11, from which we see that the coarsened classifier predicts rock classes 1, 2, and 4 with high recall values. However, it predicts rock class 3 with relatively lower recall. This class is mainly confused with rock class 1 and 4. To inspect the reasons for this, a 2D CT image cross section of the test set is compared with the coarsened CNN prediction results in Figure 7.12. This figure shows that the coarsened classifier generalizes well and predicts individual rock classes with high accuracy. The classifier even shows higher pixelwise precision in detecting thin layers and bed boundaries to the point that it is able to detect thin layers that are not picked by manual core description. As an example, in Figure 7.12, the image cross section shows a clear change in the gray scale values in the section marked by the green rectangle. Here we see that the more porous and permeable layer (characterized by darker gray scale values) is underlain by a tighter layer marked by the red rectangle. The tight layer is characterized by brighter gray scale values compared to the layers above and below, but this was not picked during manual core description. At the same time, this layer is accurately detected by the CNN classifier. More investigation of this interval reveals that the tight layer is actually a big calcite nodule encapsulated within the massive fine-grained sandstone lithofacies (Figure 17 in **paper II**). This calcite nodule is classified as rock class 1 that contains lithofacies classes with high amount of calcite cemenation, most probably due to similar gray scale values.

These results show that the ground truth may not necessarily be the ground truth as they are subjected to uncertainties, and do not show pixelwise precision. Manual core description is highly uncertain in the intervals with interchanging and dipping lithofacies, where defining a clear horizontal boundary between different lithofacies is unclear. These uncertainties can cause inconsistency and negatively affect the training process and generalization capabilities of the trained classifiers.

Table 7.4: The proposed rock classes resulted from merging similar lithofacies classes. This table is taken from **paper II**.

Rock classes	Clustered lithofacies	Description
Rock class 1	Marl, CalMarl, ContMud, WCemFineSS, Mudstone, MudsHighDens, MudFineSS, WCemBelSS, WCemMSS, CemVeryFineSS, MassVeryFineSS	Very fine to medium-grained sandstones, well cemented very fine to medium-grained sandstones, marl and mudstones
Rock class 2	GraMSSDispC, PCemGraMSS	Medium-grained granule rich sandstones, poorly cemented/ with dispersed calcite cementation
Rock class 3	SpiculiteSS, ArgFineSS	Fine-grained Spiculite sandstones and fine-grained argillaceous sandstones
Rock class 4	RippleFineSS, CrossFineSS, BioFineSS, MassFineSS, VeryFineSSHorizontal	Fine-grained sandstones with different types of laminations

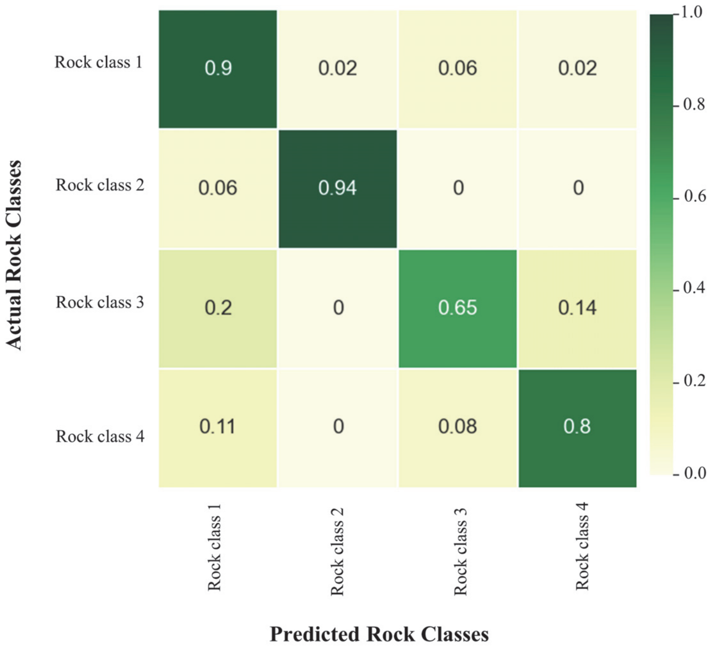


Figure 7.11: Confusion matrix on the test set using the coarsened classifier, where the original 20 lithofacies classes are merged into 4 rock classes. This figure is taken from **paper II**.

7.3 Results on lithology classification using 3D images

As described in section 6.3.2, **paper III** utilizes the 3D CT-scan sub-cubes as input to train a CNN model to classify lithofacies in an interval of the first well containing three lithofacies classes, i.e., Mudstone, GraMSSDispC, and RippleFineSS. To evaluate the generalization capability of the trained model, we used it to predict lithofacies on the unseen 3D images. More precisely, like the training set, sub-cubes of size $16 \times 16 \times 16$ pixels were extracted from the pre-processed 3D images of the test interval (i.e., nine sub-cubes per depth), and they were associated to a single lithofacies class from the manual core description. The sub-cubes were then used as input to the trained classifier to predict lithofacies.

The resulted prediction accuracy metrics and confusion matrix are presented in Table 7.5 and Figure 7.13, respectively. Here, the overall misclassification rate is 3%, which shows high generalization capability of the proposed classifier. A 2D cross section of the test interval, together with lithofacies from manual core description

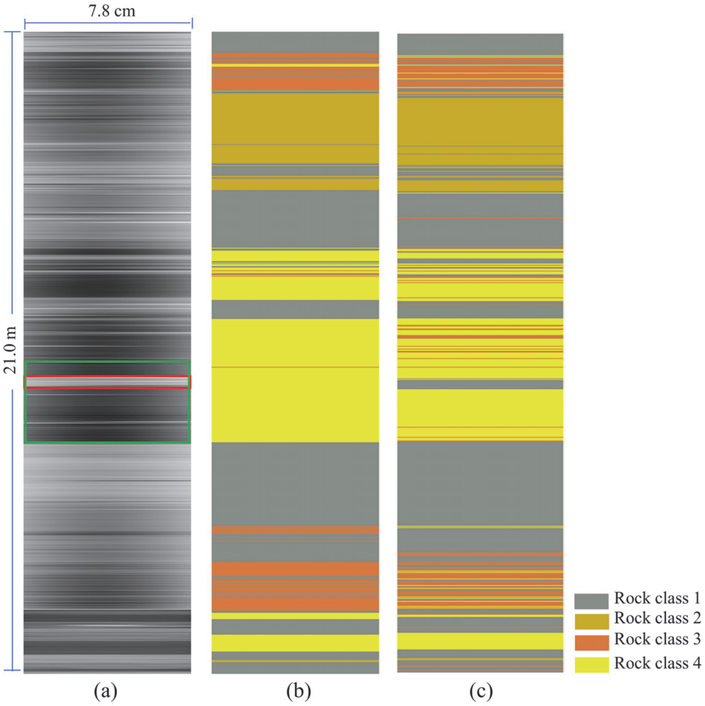


Figure 7.12: Predicted rock classes on the test section of the well (approximately 21 meters) (c), shown with actual rock classes (b) and the 2D cross section of the input CT images (a). The coarsened classifier is predicting the rock classes with high accuracy. This figure is taken from **paper II**.

and predicted results, per sub-cube, from the CNN is shown in Figure 7.14. The mean of all sub-cubes, computed using the average softmax prediction probabilities of nine sub-cubes per depth, is also presented in Figure 7.14c. One can clearly see that most of the sub-cubes are correctly classified, whereas there are a few examples of misclassifications. Most of the misclassifications occur in mudstone and ripple cross laminated sandstones.

Examples of misclassified sub-cubes, belonging to these classes, are presented in Figure 7.15, where one can immediately notice the presence of high density material such as pyrite and drilling mud invasions appearing as white patches with high attenuation values (Figure 7.15a and Figure 7.15c). Moreover, ripple cross laminated sandstone samples contain areas of low gray-level attenuation values (black color) associated with missing pixel values and dipping fractures (Figure 7.15b).

These types of features and artefacts seem to be the main reason for classification deficiency in these sub-cubes. Apparently, the images containing such artefacts are not completely removed during the image pre-processing step (section 6.1).

As shown in Figure 7.13, the degree of confusion between mudstone and ripple cross laminated sandstone classes is higher than the degree of confusion between these classes and granule rich medium-grained sandstone class. 2D cross sectional CT-scans of all classes are shown in Figure 9 in **paper III**, from where one can visually observe that the mudstone and ripple cross laminated sandstone classes share similar textures and grain sizes (very fine to fine-grained) that are markedly different than those found in the granule rich medium-grained sandstone class. These similarities can somehow explain the reason for the higher degree of confusion between these two classes. Here, we see similar behaviour as what we observed in the case of 2D images, i.e., similarities in texture, grain sizes, and gray-scale values result in higher misclassification rates.

Overall, the CNN classifier trained on the 3D images performs better on unseen images compared with the one trained on the 2D image slices. These results may indicate the added value of the full 3D information for CNN in extracting relevant features and correct identification of lithofacies classes. Note that due to computational limitations a small interval of the well containing three lithofacies classes was used to train the 3D classifier. Ideally, we should train and evaluate the 3D classifier on all the 20 lithofacies classes to ensure a fair comparison with the 2D approach.

Table 7.5: Prediction metrics on the test set using the CNN classifier trained on 3D images. Note that support shows the number of images for each class in the test set. This table is taken from **paper III**.

Lithofacies Classes	Precision	Recall	F1-score	Support
Mudstone	0.94	0.97	0.96	3600
GraMSSDispC	0.97	0.99	0.98	1332
RippleFineSS	0.99	0.96	0.97	4563

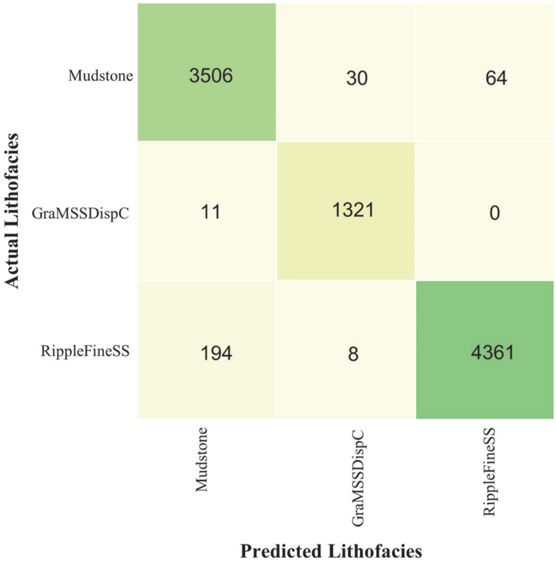


Figure 7.13: Confusion matrix on the test set using the CNN classifier trained on 3D images. This figure is taken from **paper III**.

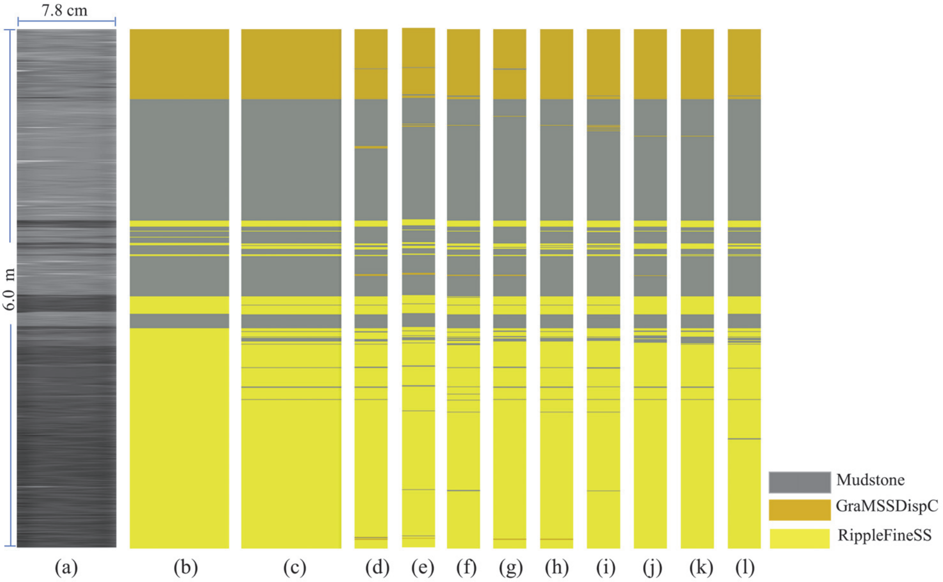
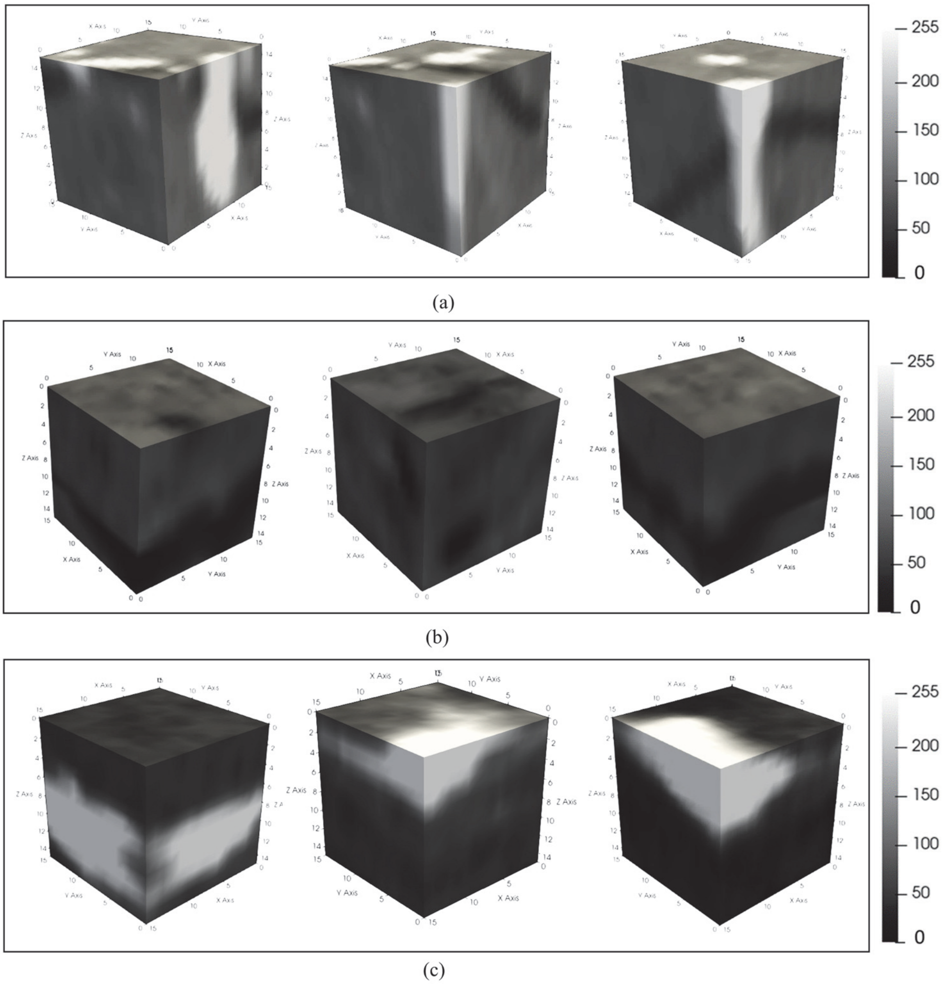


Figure 7.14: 2D cross section of the whole core CT-scan of the test interval (a) shown with actual lithofacies core description (b), mean (c) of nine 3D sub-cubes calculated using average predicted probability of nine sub-cubes at each depth and lithofacies prediction for all nine sub-cubes per depth (d-l). This figure is taken from **paper III**.



(a)

(b)

(c)

Figure 7.15: Examples of incorrectly classified mudstone (a) and ripple cross laminated fin-grained sandstone (b and c) samples by the CNN classifier trained on 3D images. Presence of high-density material with high gray-level attenuation values and missing areas with low gray-level attenuation values result in misclassification of these sub-cubes. Note that the images are coarsened by a factor of four and the size of sub-cubes is 16×16×16 pixels. This figure is taken from **paper III**.

7.4 Results on porosity estimation

As described in section 6.4, two approaches (i.e., a CNN regression model and a simple LR model) were employed to predict porosity using RCA measured porosity as the ground truth. The CNN model was trained to predict RCA porosity using the 2D image slices of the first studied well, while the LR model was trained based on the average attenuation values computed on the same 2D image slices.

7.4.1 Outlier detection results

As stated in section 6.4.3, we considered a step of outlier detection before training the LR model. The outlier detection results are presented in Figure 7.16, where the detected outliers are shown with black circles. One can observe that the data points related to the images with partly missing pixels and high density material (red and blue ellipsoids in Figure 6.11) are detected as outliers by the iForset algorithm. These data points were removed from the training set before fitting the linear regression model. The blue line in Figure 7.16 represents the fitted model.

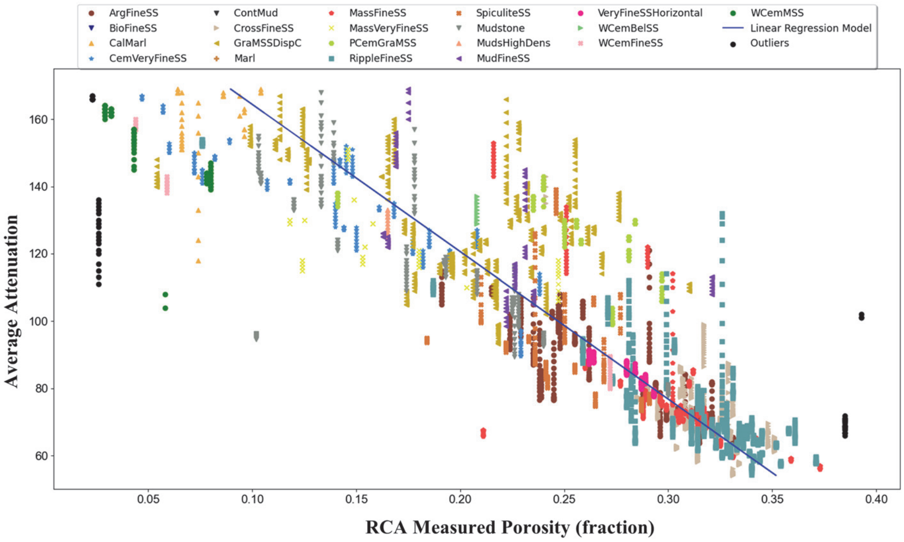


Figure 7.16: Average attenuation of 2D image slices versus RCA-derived porosity measurements used for linear regression model training. Different colors correspond to different lithofacies in the studied well. The outliers detected by iForest algorithm are shown with black circles. The regression model is trained on the data points excluding the outliers. This figure is taken from **paper IV**.

7.4.2 Training results

Figure 7.17 compares the performance of the LR and CNN models on the training data. Here, the measured porosity is plotted versus the predicted porosity in the first row, and the corresponding residual plots are shown in the second row. One can clearly see that the CNN predicted and measured porosity values are in good agreement (r^2 (coefficient of determination) and MAE of 0.997 and 0.005, respectively), while there are bigger deviations between the measured and LR predicted values (r^2 and MAE of 0.768 and 0.027, respectively). Moreover, the residual errors for the CNN model are randomly distributed around 0, while the residual errors for the LR model exhibit patterns with various degrees of scattering for various fitted porosity values.

The obtained training results indicate that average attenuation does not fully explain the porosity variations, and the distribution of pixel intensities in the images is important for an accurate porosity estimation. Moreover, this means that the pixel intensity variations that are captured by the CNN model (in the form of extracted features) translate into a learning of the relationship that exists between these variations and the porosity values. Given these training results, we continue with evaluating the generalization capabilities of the CNN model as the best performing model for porosity estimation.

To evaluate the prediction performance of the above CNN model, we used it to predict porosity on the test set (10% of labeled images in the first well) coming from the same well. Image pre-processing and labeling procedures, similar to the training set (section 6.4.1), were performed to prepare the images and labels of the test set. Note that the labels were not presented to the model; they were used for comparison purposes afterwards.

The predicted porosity on the test set is plotted versus measured porosity in Figure 7.18. Here, the predicted porosity values are shown with error bars, where the markers signify the mean predicted porosity of 19 successive images (section 6.4.1), while the error bars show the standard deviations, i.e., the variability of the predicted porosity for the 19 images. The mean predicted porosity and the actual measured porosity show an r^2 of 0.81. We can see that the error bars for most of the samples cross the line of equality (1:1 line), showing that the correlation is balanced. However, there are a few samples that occur relatively far below the equality line. These are shown within the red dashed ellipsoid in Figure 7.18.

Examples of these images are presented in Figure 7.19. The images in the first row

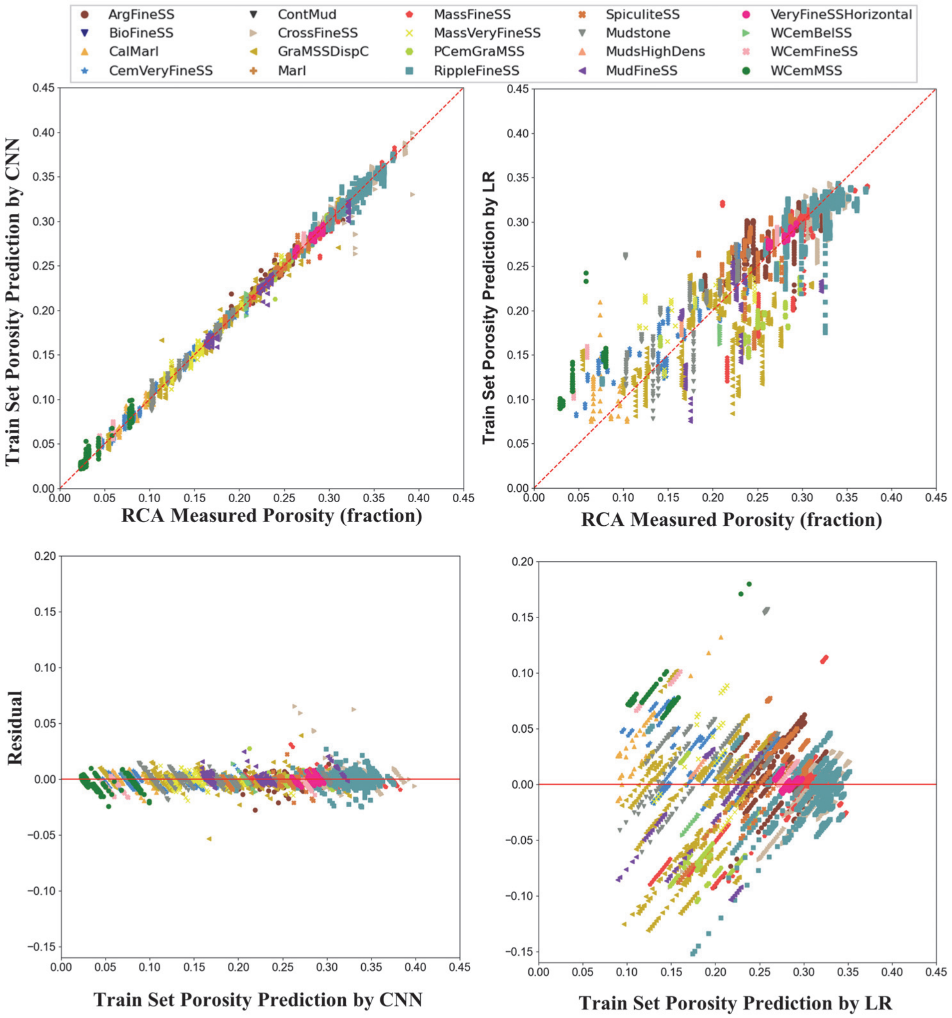


Figure 7.17: Porosity prediction of the training dataset. In the first row, the predicted porosity is plotted versus the actual measured porosity for the CNN (left) and the linear regression (right) model. The red dashed line represents the 1:1 line. The corresponding residual error plots are shown in the second row. This figure is taken from **paper IV**.

are characterized by high drilling mud invasion, where mud appears as very bright to white features with high gray-level attenuation values. The images in the second and third rows show lower gray-level attenuation values (darker) in the middle compared to the brighter edges. As mentioned before, this type of artefacts is characteristic of the image intervals with core barrel couplings. Apparently, these images are not completely removed during pre-processing, and the mentioned artefacts are probably the reason for model deficiency in porosity prediction in these images. As indicated on the images (Figure 7.19), here, the predicted porosity is lower than the measured porosity, which is expected given the presence of drilling mud and core barrel coupling features with high gray-level attenuations.

Moreover, the training dataset contains imbalanced proportions of various lithofacies classes, meaning that the CNN model is trained on fewer image examples for classes in minority compared to the more frequent ones. Given such an imbalanced training dataset, the model is expected to show lower prediction performance on the minority classes. The proportion of different lithofacies classes together with corresponding average prediction errors are presented in Table 4 of **paper IV**, where one can clearly observe higher average prediction error for lithofacies classes in minority compared to the more frequent ones. Note that high average prediction error for the PCemGraMSS class is associated with image artefacts (as explained above).

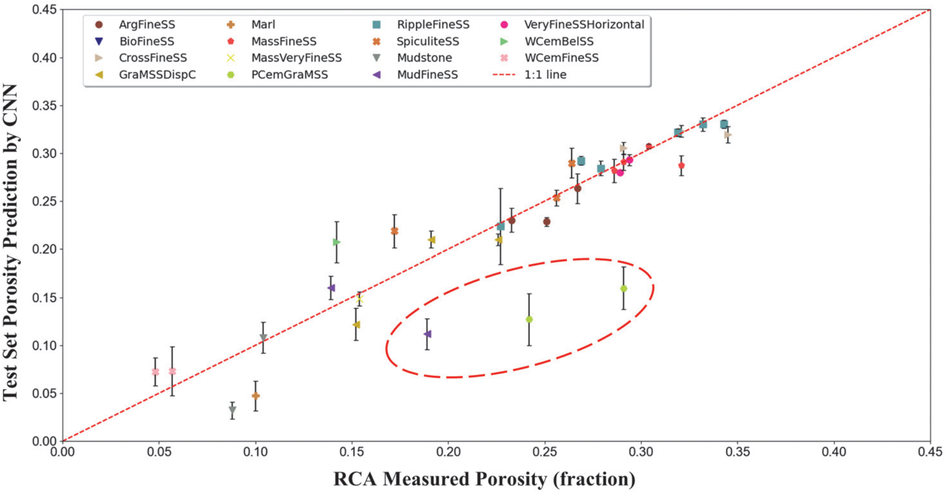


Figure 7.18: Porosity prediction of the unseen test images plotted versus the actual measured porosity. The results show an r^2 of 0.81, where high deviations occur in images with drilling mud invasion and core barrel coupling artefacts. This figure is taken from **paper IV**.

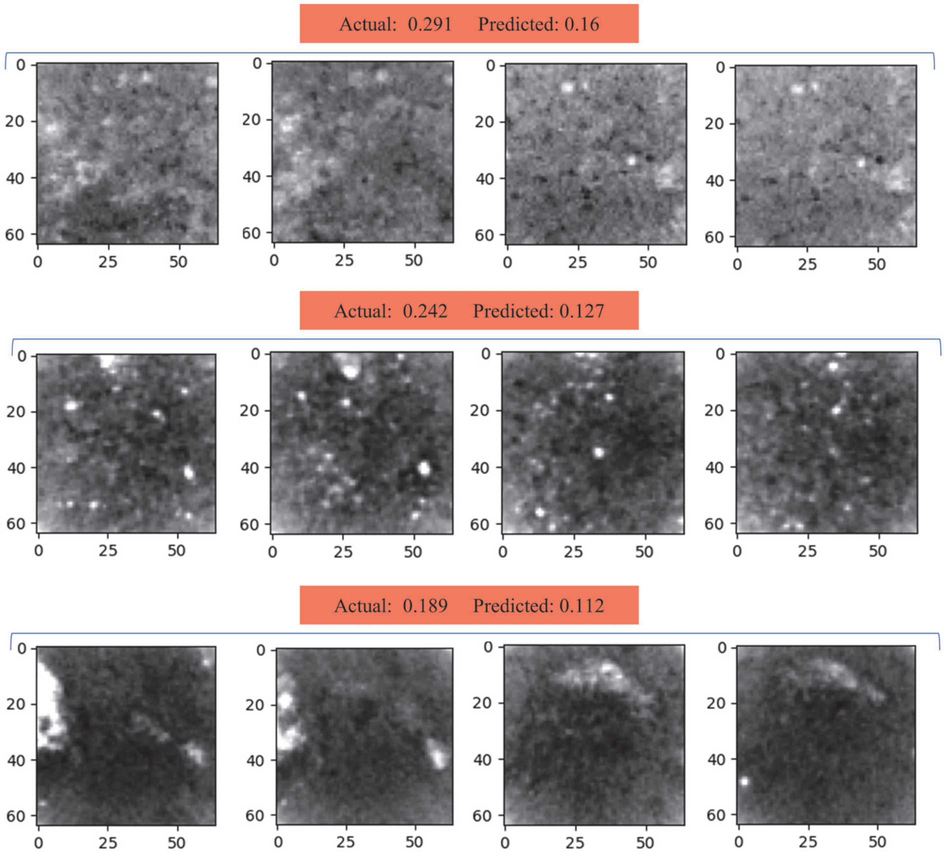


Figure 7.19: Examples of the 2D images from the test set with bigger deviations between the actual measurements and the predicted porosity values. We assume image artefacts associated with mud invasion (first row) and core barrel couplings (second and third row) are a plausible reason for the model deficiency. This figure is taken from **paper IV**.

7.4.3 Populating the studied wells with porosity values

As described above, the proposed CNN regression model for porosity prediction showed satisfying performance on the unseen images. Therefore, we employed this model to populate both wells with millimeter scale porosity values. Note that this interpolation step was performed using a model that was trained and evaluated on approximately 3% of the available images in the first well.

The acquired results for the first well are presented in Figure 7.20, where one can see a clear correlation between the predicted (light blue curve) and actual measured porosity (red circles) values. The available total porosity log (Φ_t) is also shown on the same figure (black curve). This log is computed using the density porosity equation (Eq. 2.4). For comparison of the predicted results with the total porosity log at the same scale, the predicted porosity was coarsened (green curve) to match the resolution of the porosity log. More specifically, the coarsened predicted porosity was obtained by computing a moving average of the CNN predicted porosity using a window size of 30 cm. As shown in Figure 7.20, the coarsened predicted porosity is well correlated with total porosity log. A 1.5 meter interval of the well is zoomed in and shown in Figure 7.20. Here, we can see millimeter scale porosity variations provided by the proposed CNN model, while the other two curves (i.e., total porosity log and coarsened predicted porosity) are averaged and do not show finer scale variations. Moreover, the predicted porosity is more accurate and more in line with the core porosity measurements compared to the total porosity log. Note that a bulk depth shift (section 7.1) has been applied on the CNN images and RCA porosity measurements to match the depth of the total porosity log. As this is a bulk depth shift, local depth-match issues might exist. This might give an impression of larger discrepancies, in some intervals, when comparing coarsened predicted porosity with porosity log.

The predicted porosity values for the second well are presented in Figure 7.21. As described in section 5.2, the second well penetrates only Formation 3 and Formation 4 (as shown on Figure 7.21). The lithology of Formation 4 in this well is very similar to the first well, consisting of mud and caliche rich marlstones. However, Formation 3 is quite different than the one in the first well, i.e., it contains intervals with more coarse-grained lithofacies. One can see quite good correlation between model predictions and measured porosity values, especially for depth intervals with lithology similar to the ones in the first well (i.e., depth interval above XX17). However, the model predictions are less accurate for porosity values in the intervals where the lithology is significantly different than the lithology types that the model was trained on.

Overall, the CNN model achieves an r^2 of 0.73 and MAE of 0.032. Considering that the model was trained on a single well, these results are satisfying and indicate the practical applicability of the proposed CNN model for porosity prediction on other wells.

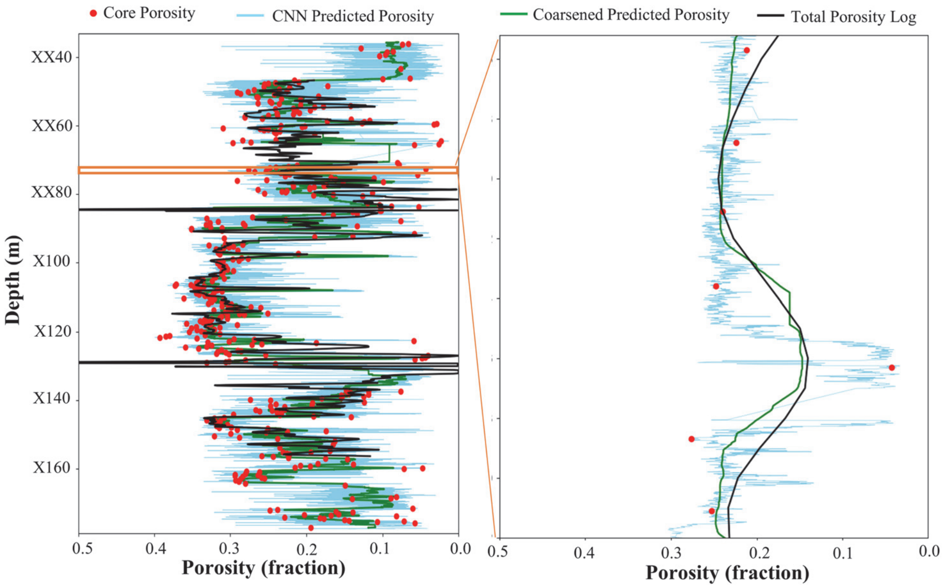


Figure 7.20: Populating the first well with porosity values. The CNN predicted porosity is in line with the RCA measured porosity. A 1.5 meter interval is zoomed in and shown in the plot to the right. This figure is taken from **paper IV**.

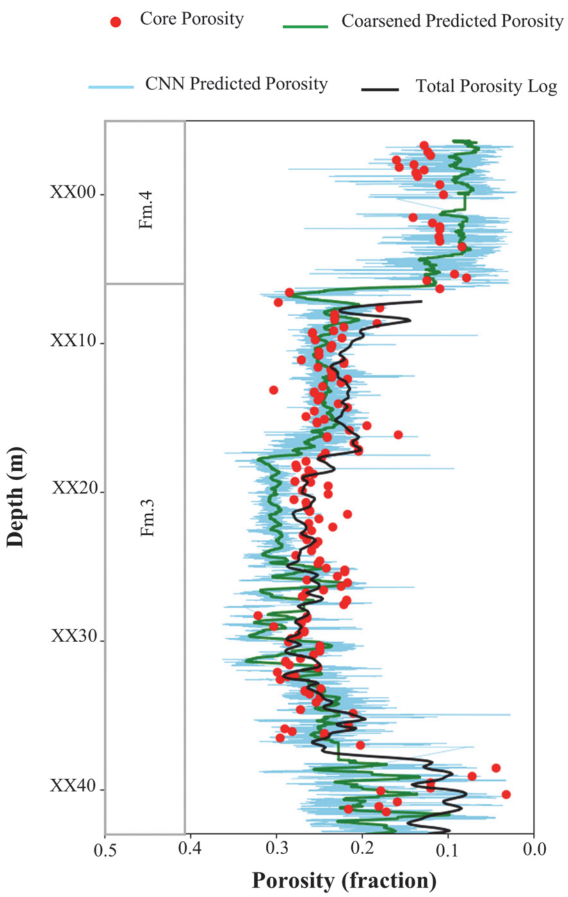
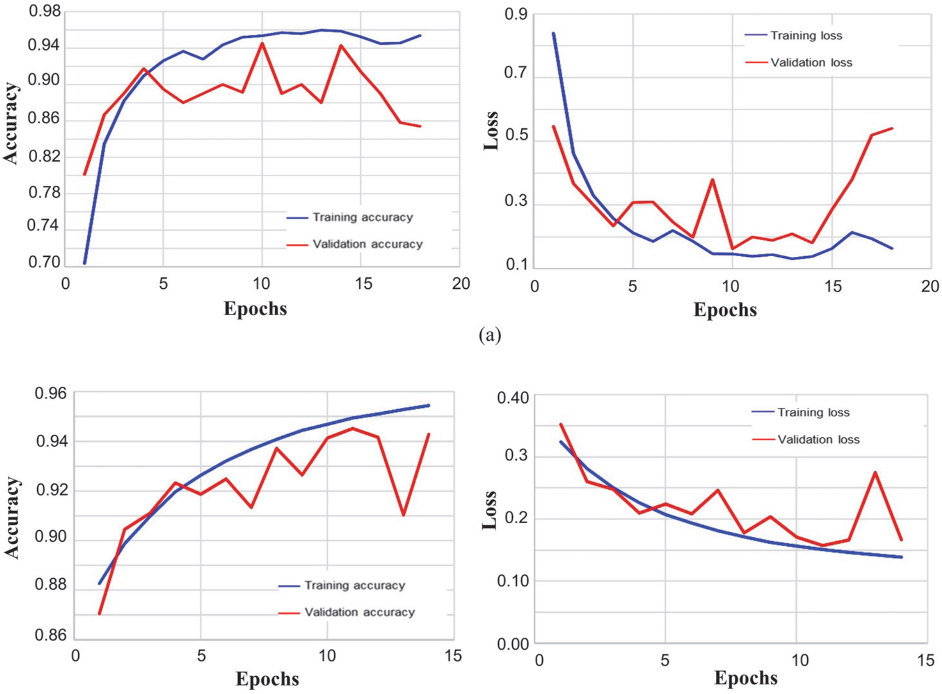


Figure 7.21: Porosity prediction in the second well employing the CNN model trained on the first well. This figure is taken from **paper IV**.

7.5 CNN architecture

As described, in **paper II** and **paper IV**, we have employed a CNN architecture with convolutional layers followed by pooling and fully connected layers. This architecture was considered as a base case architecture instead of using more complicated and deeper architectures with higher number of trainable parameters and hyperparameters. In fact, the employed CNN architectures in these two papers are similar to the AlexNet architecture (Krizhevsky et al. 2012), but with smaller number of convolutional and fully connected layers. Since AlexNet, there have been various improvements to the network architectures and feature identifications. Bearing this in mind, we also considered the VGG16 architecture (Simonyan and Zisserman 2014) as a deeper architecture to classify lithofacies classes using 2D image slices.

Here, we considered two approaches; in the first approach the original VGG16 architecture was used without any kind of hyperparameter tuning. The VGG16 model was trained with a learning rate of 0.0001 using Adam optimizer. The performance of this model on the training and validation sets is presented in Figure 7.22a, where one can see that the training process stops at approximately 0.957 accuracy and the trained model does not generalize well on the validation set. In a second approach, we tried to tune the hyperparameters using Keras Tuner. The considered hyperparameters were number of kernels in the convolutional layers, number of neurons in the hidden layers, learning rate, and dropout rate. However, using batch sizes of 32, we were running out of GPU memory using available GPU resources (1 GPU, *NVIDIA GeForce GTX 1050* graphic card). Therefore, we used a batch size of 16 instead of 32. The performance of the best model configuration on the training and validation sets is shown in Figure 7.22b. This model was then used to predict the lithofacies classes on the test set. The accuracy metrics and the confusion matrix of the tuned VGG16 are presented in Table 7.6 and Figure 7.23, respectively. This model shows an overall accuracy of 0.55, while the proposed classifier in **paper II** performs slightly better with an overall accuracy of 0.56. The results are comparable. However, overall, the proposed model performs better. In addition, it requires lower computational time and resources given its lower complexity.



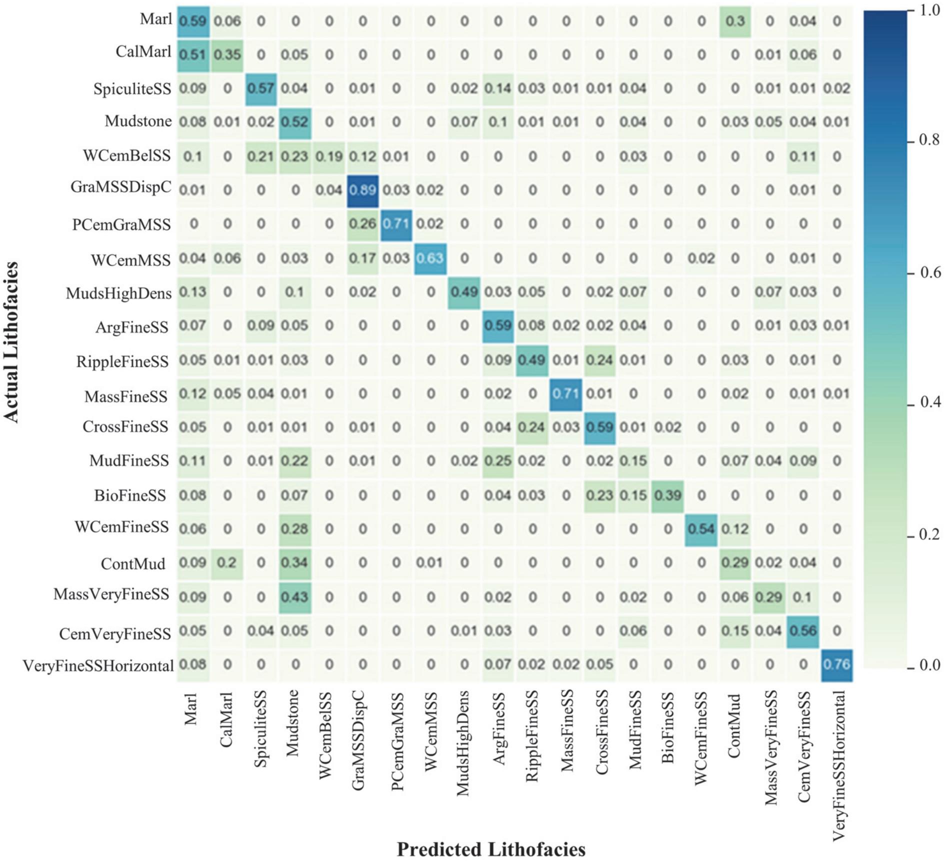
(a)

(b)

Figure 7.22: The performance of original VGG16 (a) and tuned VGG16 (b) on the training and validation sets.

Table 7.6: Prediction accuracy metrics on the test set using the tuned VGG16 architecture. Support shows the number of predicted samples for each class.

Lithofacies labels	Precision	Recall	F1-score	Support
Marl	0.08	0.59	0.15	542
CalMarl	0.24	0.35	0.28	918
SpiculiteSS	0.56	0.57	0.56	1835
Mudstone	0.50	0.52	0.51	6684
WCemBelSS	0.13	0.19	0.16	160
GraMSSDispC	0.84	0.89	0.86	4498
PCemGraMSS	0.86	0.71	0.78	1491
WCemMSS	0.80	0.63	0.70	1161
MudsHighDens	0.13	0.49	0.20	187
ArgFineSS	0.50	0.59	0.54	3774
RippleFineSS	0.45	0.49	0.47	2879
MassFineSS	0.88	0.71	0.79	3522
CrossFineSS	0.73	0.59	0.65	5096
MudFineSS	0.24	0.15	0.18	1979
BioFineSS	0.74	0.39	0.51	824
WCemFineSS	0.82	0.54	0.65	653
ContMud	0.45	0.29	0.35	3489
MassVeryFineSS	0.47	0.29	0.36	2121
CemVeryFineSS	0.59	0.56	0.57	2906
VeryFineSSHorizontal	0.80	0.76	0.78	1192



	Marl	CalMarl	SpiculiteSS	Mudstone	WCemBelSS	GraMSSDispC	PCemGraMSS	WCemMSS	MudsHighDens	ArgFineSS	RippleFineSS	MassFineSS	CrossFineSS	MudFineSS	BioFineSS	WCemFineSS	ContMud	MassVeryFineSS	CemVeryFineSS	VeryFineSSHorizontal
Marl	0.59	0.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0.04	0
CalMarl	0.51	0.35	0	0.05	0	0	0	0	0	0	0	0	0	0	0	0	0	0.01	0.06	0
SpiculiteSS	0.09	0	0.57	0.04	0	0.01	0	0	0.02	0.14	0.03	0.01	0.01	0.04	0	0	0	0.01	0.01	0.02
Mudstone	0.08	0.01	0.02	0.52	0	0.01	0	0	0.07	0.1	0.01	0.01	0	0.04	0	0	0.03	0.05	0.04	0.01
WCemBelSS	0.1	0	0.21	0.23	0.19	0.12	0.01	0	0	0	0	0	0	0.03	0	0	0	0	0.11	0
GraMSSDispC	0.01	0	0	0	0.04	0.89	0.03	0.02	0	0	0	0	0	0	0	0	0	0	0.01	0
PCemGraMSS	0	0	0	0	0	0.26	0.71	0.02	0	0	0	0	0	0	0	0	0	0	0	0
WCemMSS	0.04	0.06	0	0.03	0	0.17	0.03	0.63	0	0	0	0	0	0	0	0.02	0	0	0.01	0
MudsHighDens	0.13	0	0	0.1	0	0.02	0	0	0.49	0.03	0.05	0	0.02	0.07	0	0	0	0.07	0.03	0
ArgFineSS	0.07	0	0.09	0.05	0	0	0	0	0	0.59	0.08	0.02	0.02	0.04	0	0	0	0.01	0.03	0.01
RippleFineSS	0.05	0.01	0.01	0.03	0	0	0	0	0	0.09	0.49	0.01	0.24	0.01	0	0	0.03	0	0.01	0
MassFineSS	0.12	0.05	0.04	0.01	0	0	0	0	0	0.02	0	0.71	0.01	0	0	0	0.02	0	0.01	0.01
CrossFineSS	0.05	0	0.01	0.01	0	0.01	0	0	0	0.04	0.24	0.03	0.59	0.01	0.02	0	0	0	0	0
MudFineSS	0.11	0	0.01	0.22	0	0.01	0	0	0.02	0.25	0.02	0	0.02	0.15	0	0	0.07	0.04	0.09	0
BioFineSS	0.08	0	0	0.07	0	0	0	0	0	0.04	0.03	0	0.23	0.15	0.39	0	0	0	0	0
WCemFineSS	0.06	0	0	0.28	0	0	0	0	0	0	0	0	0	0	0	0.54	0.12	0	0	0
ContMud	0.09	0.2	0	0.34	0	0	0	0.01	0	0	0	0	0	0	0	0	0.29	0.02	0.04	0
MassVeryFineSS	0.09	0	0	0.43	0	0	0	0	0	0.02	0	0	0	0.02	0	0	0.06	0.29	0.1	0
CemVeryFineSS	0.05	0	0.04	0.05	0	0	0	0	0.01	0.03	0	0	0	0.06	0	0	0.15	0.04	0.56	0
VeryFineSSHorizontal	0.08	0	0	0	0	0	0	0	0	0.07	0.02	0.02	0.05	0	0	0	0	0	0	0.76

Figure 7.23: Prediction confusion matrix of the test set using the tuned VGG16 model.

7.6 Division of data between training and validation

In **paper II** and **paper IV**, 2D image slices of the train and validation sets were divided randomly, whereas the images of the test set were separated manually using continuous intervals of different lithofacies classes taken from various depth intervals. In addition, in **paper II**, an image augmentation process was performed before splitting data into the training and validation sets. Note that this was not the case in **paper IV**, where augmentation of the training images were performed after the train-validation split.

As the lithofacies classes are varying gradually, a random split of the 2D image slices can result in similar images in the train and validation sets. Moreover, image augmentation prior to splitting may result in images in the validation set almost identical to the images of the train set. Therefore, evaluating the trained model on such a validation set might result in overestimation of the performance of the model. As seen in **paper II** (Figure 8), the trained classifier apparently performs very well on the validation set, while the performance on the test set is deteriorated.

To evaluate the effect of the presence of augmented images in the validation set, we randomly separated the train and validation sets prior to image augmentation. Then, the training images were augmented and the best model configuration was validated on this new set of non-augmented images using the same training process as explained in section 6.3.2. The model performance during training is compared to the performance of the model of **paper II** in Figure 7.24. Investigating these figures, one can see that these two models show similar performance on the validation sets with and without augmented images.

To evaluate the effect of randomness, we manually separated the validation set using continuous intervals of lithofacies classes. Here, we used the same train-validation splitting ratio as in **paper II** (i.e., approximately 80% for training and 20% for validation). The distribution of various lithofacies classes in the resulting train and validation sets are presented in Figure 7.25. Then, the best model configuration was derived from a hyperparameter tuning process, this time using the mentioned manual train and validation split.

The performance of the best model configuration on the training and validation sets is presented in Figure 7.26, where one can see that the model is not able to generalize well on the validation set and the overall accuracy on the validation set stays around 0.56. The training process stops at an accuracy of approximately 0.78 after 20 epochs. This behaviour is completely opposite of what we saw in the

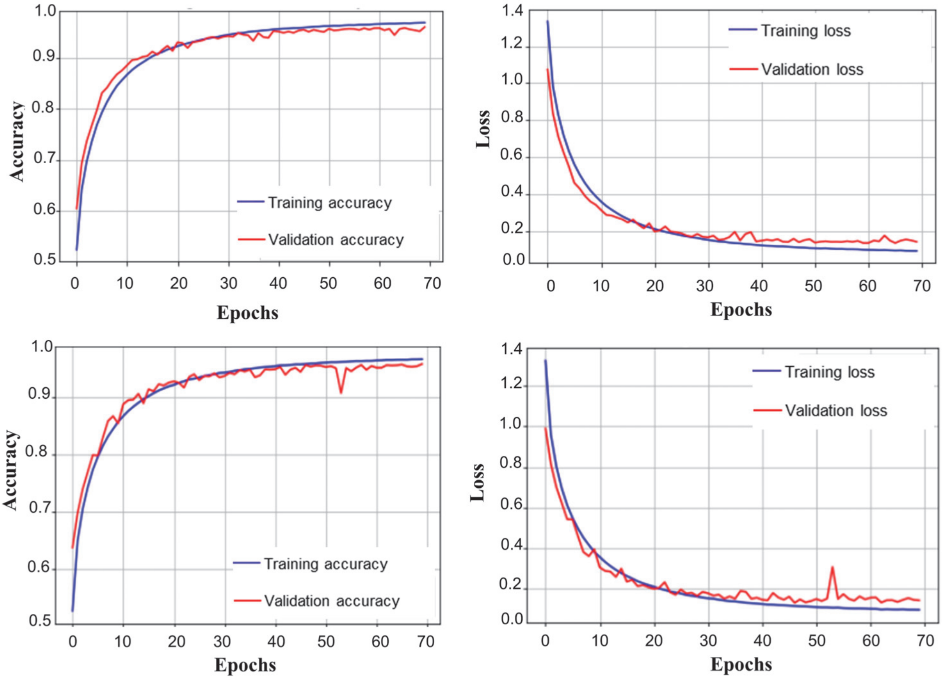


Figure 7.24: The CNN Model performance on the training and validation sets with (top) and without (bottom) augmented images.

case of random train-validation split in **paper II** (Figure 8), where both accuracy and loss of the validation set improved with increasing the training time and the training process continued up to 70 epochs.

To further investigate these results, we trained the classifier of **paper II** for only 20 epochs and used it to predict the lithofacies classes of the test set. Moreover, the model validated on the validation set with manual split was also employed to predict the lithofacies classes on the same test set. The obtained prediction results are compared in Table 7.7. Note that both models obtained an overall prediction accuracy of 0.55 and in most of the lithofacies classes, these two models show similar performance, although, one model might perform better on a specific class compared to the other.

These results indicate that a random split of the train and validation sets, in case of 2D image slices with gradual changes, can be misleading and result in overestimation of the performance of the trained model on the validation set. In fact, the prediction results on the unseen test images give a more realistic performance of the trained classifier. More importantly, the worse performance of the model on the manually separated validation set might be an indication of overfitting due to lack of enough training images. Therefore, the performance of the model might improve by including more images from other wells with similar lithology. Another interesting experiment, in the absence of more images, would be to consider transfer learning where the existing images of this single well can be used to further train a model previously trained on huge number of different, but related images.

All that said, the similarities in rock properties of the considered 20 lithofacies classes might still be the reason for high misclassification rates and lower overall accuracy on the validation and test sets. In **paper II**, we observed that merging similar lithofacies classes into rock classes improved the prediction results.

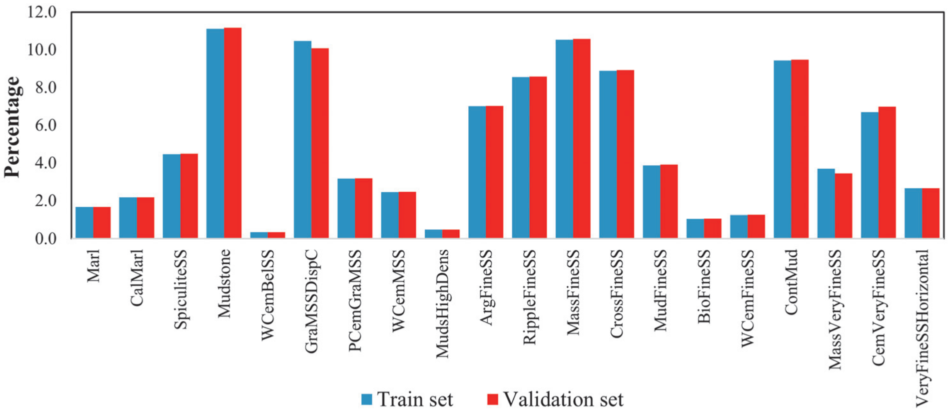


Figure 7.25: Distribution of different lithofacies classes in the train and validation sets separated manually using continuous intervals of lithofacies classes.

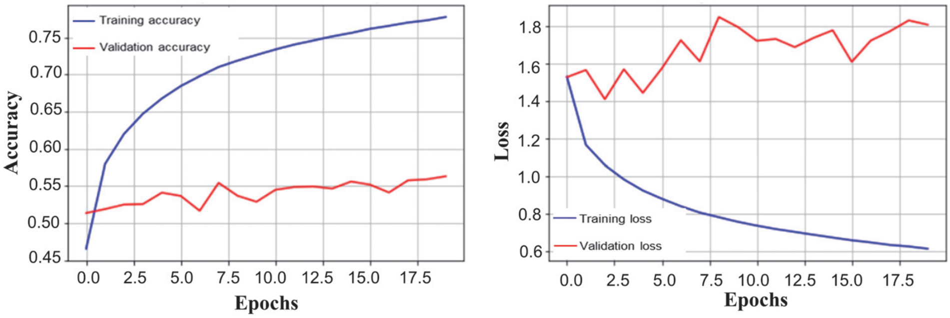


Figure 7.26: Performance of the trained model on the validation set acquired manually. The plot to the left shows the accuracy results by increasing the number of epochs, whereas the plot to the right shows loss results by increasing the number of epochs.

Table 7.7: Comparison of prediction metrics of the CNN classifiers with random (Rand) and manual (Man) train-validation split. Note that in case of manual split, the validation set was selected manually using continuous 2D image slices of each lithofacies class taken from varous intervals. Both models were trained for 20 epochs.

	Precision		Recall		F1-score	
Lithofacies Labels	Rand	Man	Rand	Man	Rand	Man
Marl	0.15	0.04	0.30	0.04	0.20	0.04
CalMarl	0.26	0.42	0.45	0.84	0.33	0.56
SpiculiteSS	43	0.48	0.70	0.56	0.54	0.52
Mudstone	0.43	0.45	0.53	0.64	0.55	0.53
WCemBelSS	0.17	0.14	0.17	0.10	0.17	0.12
GraMSSDispC	0.85	0.85	0.90	0.90	0.87	0.88
PCemGraMSS	0.80	0.86	0.80	0.78	0.80	0.82
WCemMSS	0.85	0.72	0.53	0.71	0.66	0.71
MudsHighDens	0.27	0.18	0.41	0.22	0.32	0.20
ArgFineSS	0.49	0.39	0.53	0.51	0.51	0.44
RippleFineSS	0.35	0.31	0.63	0.30	0.45	0.30
MassFineSS	0.87	0.87	0.73	0.74	0.79	0.80
CrossFineSS	0.69	0.70	0.35	0.49	0.47	0.57
MudFineSS	0.29	0.27	0.29	0.12	0.29	0.17
BioFineSS	0.80	0.80	0.33	0.37	0.47	0.50
WCemFineSS	0.62	0.64	0.63	0.91	0.63	0.75
ContMud	0.44	0.48	0.37	0.33	0.40	0.39
MassVeryFineSS	0.41	0.35	0.34	0.24	0.37	0.28
CemVeryFineSS	0.50	0.66	0.54	0.50	0.52	0.57
VeryFineSSHorizontal	0.83	0.64	0.74	0.78	0.78	0.70

Chapter 8. Conclusions

The main question investigated in this thesis was to understand whether the information stored in whole core CT-scans is sufficient to automatically classify different lithofacies and to estimate transport properties. To investigate this, four approaches were considered as presented in **papers I**, **II**, **III**, and **IV**.

Overall, the obtained lithofacies classification results based on 2D CT images (**papers I** and **II**) revealed that CNN classifier performs better on predicting unseen images compared to SVM classifier. However, we should consider that these two approaches use different types of data to classify lithofacies. Therefore, apart from the inherent differences in the statistical generalization capabilities of these algorithms in general, there are other factors that may have affected the performance of these algorithms in the considered workflows. The best SVM classifier found during the PhD project was trained based on specific statistical and textural features extracted from 2D cross-sectional images. This type of images, and the features extracted from them, might not be completely representative of the 3D variations in core data. Depending on the type of core material, the availability of full 3D data might be necessary to distinguish lithofacies classes.

Both classifiers (CNN and SVM) showed lithofacies-dependent accuracy, i.e., specific lithofacies classes were detected with satisfying performance, while other classes were misclassified to various degrees. Apparently, lithofacies that share similarities in the texture and grayscale attenuation values are confused by the classifiers. More importantly, additional analyses of the porosity-permeability trends indicate that misclassified lithofacies share similar transport properties. Therefore, the information acquired from prediction results can help in understanding the similarities between various lithofacies classes and their corresponding transport properties. In **paper II** these results were used as a tool to cluster similar lithofacies classes into coarser rock classes as a post-classification step to refine the acquired results.

The CNN classifier trained on the 3D CT images generalizes better compared with the one trained on the 2D image slices, achieving an overall prediction accuracy of 97% (The CNN model trained on 2D images achieved an accuracy of 56%). These results may indicate the added value of the full 3D information for CNN in extracting relevant features and correct identification of lithofacies classes. However, we should note that, due to computational limitations, only a small interval of the well consisting of three (out of twenty) lithofacies classes was used to train the 3D classifier. Ideally, we should train and evaluate the 3D classifier on all the lithofacies classes to ensure a fair comparison with the 2D approach.

All the proposed classifiers show higher pixelwise precisions compared to the expert-derived core descriptions, providing higher resolution information in detecting thin layers and fine scale heterogeneities.

The capability of convolutional neural networks to automatically predict millimeter scale porosity from 2D image slices was then investigated in **paper IV**, for which RCA-derived porosity measurements were used as the training ground truth. Overall, the proposed method provides continuous millimeter scale porosity values, while such high resolution porosity variations are not detectable by the total porosity log due to its lower resolution. Our conclusion is that the proposed method can be employed to estimate continuous core scale porosity values in an automatic fashion, and at early stages of the reservoir characterization process. This method can moreover be used to calibrate the porosity logs, thereby reducing the uncertainties associated with indirect calculations of the porosity from such logs. Our developed method may also be helpful to identify core plug locations for improved core analysis.

With respect to the model training and prediction deficiencies, there are several issues to be considered:

- Most of the machine learning algorithms used for classification purposes assume an equal number of training examples for each class. In the case of lithofacies classification, since the thickness of different lithofacies varies, the distribution of classes in the training dataset is not equal, resulting in a noticeably imbalanced dataset. This can result in poor predictive performance for lithofacies classes with fewer images in the training set.

- Inappropriate separation of training and test datasets can result in model deficiencies too. Most data science analyses are well served by a random train/test split. In the case of stratified sedimentary rocks, however, the lithofacies are gradually and slowly varying. Here, a random split of CT images would result in similar images in the training and test sets, and overestimation of the performance of the model. In addition, the training dataset must adequately represent the actual variance to make sure that the trained model produces outputs that can deal with the entire dataset. The problem of poor data curation is aggravated for neural network and deep learning algorithms due to their "black box" nature, which prevents understanding the reason why a certain output is produced. Therefore, special care (coupled with domain knowledge) is required to build more representative splits and train models with good statistical generalization capabilities.

- The quality of images and the useful features that can be extracted from these images directly affect the generalization capabilities of the data-driven models built from them. Our investigations show that images with remaining undesired artefacts and specific features (e.g., images with missing pixels, high density material, and core barrel coupling) that have not been completely removed by image pre-processing steps, quickly lead to model deficiencies and incorrect prediction performances. Therefore, it is extremely important to remove images with artefacts before feature extraction or feeding them into the model since the statistical degradation associated to these issues is worse than what initially thought it would be.

- The whole core CT images are much more complex compared to the images where the considered algorithms have been successfully applied before. The PCA results (**paper I**) highlight the complexity and non-linear nature of the studied images. Moreover, images belonging to specific lithofacies classes do not contain clear features, which makes it difficult for the CNN model to distinguish them from other classes.

- In addition to the aforementioned points, the quality of manual core descriptions, i.e., the quality of the ground truth used in the proposed lithofacies classification algorithms, has a very strong effect on the training process and generalization capability of the trained classifiers. The manual core descriptions may not necessarily be the ground truth, as they are subjected to uncertainties and do not show pixelwise precision, especially in the intervals with interchanging and dipping lithofacies. Assigning a clear boundary in these intervals, during manual core description, becomes challenging and uncertain.

- With all that said, it seems that, out of all described issues, the proposed

> models are very sensitive to noise, and the image artefacts are the most influencing factors that can result in model deficiencies. Therefore, we think that the industry must put special effort into image cleaning and image preprocessing before starting the training process.

With regards to applicability of the developed methods for the industry, we think that the acquired results confirm the practicality of whole core CT-scans (as digital representations of the cored intervals) and machine learning algorithms, especially CNN, in automated lithology classification and automated porosity estimations. The introduced methodologies for lithology classification can improve the speed of lithology classification in the cored well intervals with similar lithology as the ones that the models were trained on.

Lithology classification based on extracted 3D CT sub-cubes reveals very promising results on the unseen images. Using sub-cubes instead of full three-dimensional images results in a higher amount of training images; therefore, a smaller interval of the well is needed for model training. The trained model can be used to predict the lithology in the remaining parts of the same well and even other wells in fields with similar lithology, thereby reducing manual labor and speeding up the classification process.

Moreover, the proposed CNN regression model, for porosity estimation, can provide continuous millimeter scale porosity estimations at early stages of reservoir characterization process. As stated before, the predicted porosity results can be used to calibrate the porosity logs, and reduce uncertainties associated with indirect porosity calculations. Therefore, we believe that the most significant application of the proposed porosity estimation method will be in reservoir characterization and modeling.

We remark that in this study the models were trained only on a single well. Ideally, training the models on multiple wells, covering a range of various lithological intervals and porosity values, is expected to result in more robust models with higher generalization capabilities on unseen wells.

Bibliography

Abashkin V, Seleznev I, Chertova A, Samokhvalov A, Istomin S, Romanov D (2020) Digital analysis of the whole core photos. In: First EAGE Digitalization Conference and Exhibition, European Association of Geoscientists & Engineers, vol 2020, pp 1–5

Agarwal A, Laronga R, Walker L (2013) Rotary sidewall coring—size matters. Oilfield Review 2014(25):4

Ahmadi MA, Chen Z (2019) Comparison of machine learning methods for estimating permeability and porosity of oil reservoirs via petro-physical logs. Petroleum 5(3):271–284, https://doi.org/10.1016/j.petlm.2018.06.002

Ahmadi MA, Ahmadi MR, Hosseini SM, Ebadi M (2014) Connectionist model predicts the porosity and permeability of petroleum reservoirs by means of petro-physical logs: application of artificial intelligence. Journal of Petroleum Science and Engineering 123:183–200, https://doi.org/10.1016/j.petrol.2014.08.026

Aizerman MA (1964) Theoretical foundations of the potential function method in pattern recognition learning. Automation and remote control 25:821–837

Al-Anazi A, Gates I (2010) A support vector machine algorithm to classify lithofacies and model permeability in heterogeneous reservoirs. Engineering Geology 114(3-4):267–277, https://doi.org/10.1016/j.enggeo.2010.05.005

Al-Bulushi N, King PR, Blunt MJ, Kraaijveld M (2009) Development of artificial neural network models for predicting water saturation and fluid distribution. Journal of Petroleum Science and engineering 68(3-4):197–208, https://doi.org/10.1016/j.petrol.2009.06.017

Al-Obaidi M, Heidari Z, Casey B, Williams R, Spath J (2018) Automatic well-log-based fabric-oriented rock classification for optimizing landing spots and completion intervals in the midland basin. In: SPWLA 59th Annual Logging Symposium, Society of Petrophysicists and Well-Log Analysts

Al-Saddique M, Hamada G, Al-Awad MN (2000) State of the art: Review of coring and core analysis technology. Journal of King Saud University-Engineering Sciences 12(1):117–137

Alexandris N, Gupta S, Koutsias N (2017) Remote sensing of burned areas via pca, part 1; centering, scaling and evd vs svd. Open Geospatial Data, Software and Standards 2(1):1–11, https://doi.org/10.1186/s40965-017-0028-1

Alqahtani N, Armstrong RT, Mostaghimi P (2018) Deep learning convolutional neural networks to predict porous media properties. In: SPE Asia Pacific oil and gas conference and exhibition, Society of Petroleum Engineers, https://doi.org/10.2118/191906-MS

Anjos CE, Avila MR, Vasconcelos AG, Neta AMP, Medeiros LC, Evsukoff AG, Surmas R, Landau L (2021) Deep learning for lithological classification of carbonate rock micro-ct images. Computational Geosciences pp 1–13

Ashena R, Thonhauser G (2018) Fundamental coring methods. In: Coring Methods and Systems, Springer, pp 17–22

Asquith GB, Krygowski D, Gibson CR (2004) Basic well log analysis, vol 16. American Association of Petroleum Geologists Tulsa

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *The Journal of Machine Learning Research* 13(1):281–305

Bertozzi W, Ellis D, Wahl J (1981) The physical foundation of formation lithology logging with gamma rays. Geophysics 46(10):1439–1455

Bezdek JC, Ehrlich R, Full W (1984) Fcm: The fuzzy c-means clustering algorithm. Computers & geosciences 10(2-3):191–203

Bize-Forest N, Lima L, Baines V, Boyd A, Abbots F, Barnett A (2018) Using Machine-Learning for Depositional Facies Prediction in a Complex Carbonate Reservoir. Society of Petrophysicists and Well-Log Analysts

Bordignon F, Figueiredo L, Exterkoetter R, Rodrigues BB, Correia M (2019) Deep learning for grain size and porosity distributions estimation on micro-ct images. In: Proceedings of the 16th International Congress of the Brazilian Geophysical Society & Expogef, https://doi.org/10.22564/16cisbgf2019.209

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, pp 144–152

Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery 2(2):121–167

Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. SIAM Journal on scientific computing 16(5):1190–1208

Chang Hc, Kopaska-Merkel DC, Chen HC, Durrans SR (2000) Lithofacies identification using multiple adaptive resonance theory neural networks and group decision expert system. Computers & Geosciences 26(5):591–601

Cheng G, Guo W (2017) Rock images classification by using deep convolution neural network. In: Journal of Physics: Conference Series, IOP Publishing, p 012089

Choi D, Shallue CJ, Nado Z, Lee J, Maddison CJ, Dahl GE (2019) On empirical comparisons of optimizers for deep learning. arXiv preprint arXiv:191005446

Chollet F, et al. (2015) Keras. URL `https://github.com/fchollet/keras`

Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20(3):273–297

Darling T (2005) Well logging and formation evaluation. Elsevier

Dubois MK, Bohling GC, Chakrabarti S (2007) Comparison of four approaches to a rock facies classification problem. Computers & Geosciences 33(5):599–617, https://doi.org/10.1016/j.cageo.2006.08.011

Dunham MW, Malcolm A, Welford JK (2020) Improved well log classification using semisupervised gaussian mixture models and a new hyper-parameter selection strategy. Computers & Geosciences 140:104501, https://doi.org/10.1016/j.cageo.2020.104501

Elkatatny S, Mahmoud M, Tariq Z, Abdulraheem A (2018) New insights into the prediction of heterogeneous carbonate reservoir permeability from well logs using artificial intelligence network. Neural Computing and Applications 30(9):2673–2683, https://doi.org/10.1007/s00521-017-2850-x

Ellis DV, Singer JM (2007) Well logging for earth scientists, vol 692. Springer

Ershaghi I, Ershaghi MA, Popa A (2018) Data ethics in oil and gas operations. In: SPE Western Regional Meeting, Society of Petroleum Engineers, https://doi.org/10.2118/190117-MS

Evgeniou T, Pontil M (1999) Support vector machines: Theory and applications. In: Advanced Course on Artificial Intelligence, Springer, pp 249–257

Gardner JS, Dumanoir J (1980) Litho-density log interpretation. In: SPWLA 21st Annual Logging Symposium, Society of Petrophysicists and Well-Log Analysts

Godbole S (2002) Exploiting confusion matrices for automatic generation of topic hierarchies and scaling up multi-way classifiers

Goldman LW (2007) Principles of ct and ct technology. Journal of nuclear medicine technology 35(3):115–128

Gonzalez A, Kanyan L, Heidari Z, Lopez O (2019) Integrated multi-physics workflow for automatic rock classification and formation evaluation using multi-scale image analysis and conventional well logs. In: SPWLA 60th Annual Logging Symposium, Society of Petrophysicists and Well-Log Analysts, https://doi.org/10.30632/T60ALS-2019_A

Gonzalez A, Heidari Z, Lopez O (2020) Integration of 3d volumetric ct-scan image data with conventional well logs for detection of petrophysical rock classes. In: SPWLA 61st Annual Logging Symposium, OnePetro, https://doi.org/10.30632/SPWLA-5071

Goupillaud P, Grossmann A, Morlet J (1984) Cycle-octave and related transforms in seismic signal analysis. Geoexploration 23(1):85–102, https://doi.org/10.1016/0016-7142(84)90025-5

Hall B (2016) Facies Classification Using Machine Learning. The Leading Edge 35(10):906–909, https://doi.org/10.1190/tle35100906.1

Hall B, Govert A (2016) Techniques for Using Core CT Data for Facies Identification and Analysis. Unconventional Resources Technology ConferenceAt: San Antonio, Texas

Hall M, Hall B (2017) Distributed collaborative prediction: Results of the machine learning contest. The Leading Edge 36(3):267–269

Hanson S, Pratt L (1988) Comparing biases for minimal network construction with back-propagation. Advances in neural information processing systems 1:177–185

Haralick RM, Shanmugam K, Dinstein IH (1973) Textural features for image classification. IEEE Transactions on systems, man, and cybernetics 6:610–621, https://doi.org/10.1109/TSMC.1973.4309314

Hartmann DJ, Beaumont EA (1999) Treatise of Petroleum Geology/Handbook of Petroleum Geology: Exploring for Oil and Gas Traps. Chapter 9: Predicting Reservoir System Quality and Performance. AAPG Special Volumes

Hébert V, Porcher T, Planes V, Léger M, Alperovich A, Goldluecke B, Rodriguez O, Youssef S (2020) Digital core repository coupled with machine learning as a tool to classify and assess petrophysical rock properties. In: E3S Web of Conferences, EDP Sciences, vol 146, p 01003, https://doi.org/10.1051/e3sconf/202014601003

Helle HB, Bhatt A, Ursin B (2001) Porosity and permeability prediction from wireline logs using artificial neural networks: a north sea case study. Geophysical Prospecting 49(4):431–444, https://doi.org/10.1046/j.1365-2478.2001.00271.x

Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:12070580

Horrocks T, Holden EJ, Wedge D (2015) Evaluation of automated lithology classification architectures using highly-sampled wireline logs for coal exploration. Computers & geosciences 83:209–218, https://doi.org/10.1016/j.cageo.2015.07.013

Hutter F, Hoos HH, Leyton-Brown K (2011) Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization, Springer, pp 507–523

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167

Jungmann M, Kopal M, Clauser C, Berlage T (2011) Multi-class supervised classification of electrical borehole wall images using texture features. Computers & Geosciences 37(4):541–553, https://doi.org/10.1016/j.cageo.2010.08.008

Ketcham RA, Carlson WD (2001) Acquisition, optimization and interpretation of X-ray computed tomographic imagery: applications to the geosciences. Computers & Geosciences 27(4):381–400, https://doi.org/10.1016/S0098-3004(00)00116-3

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25:1097–1105

LeCun Y, et al. (1989) Generalization and network design strategies. Connectionism in perspective 19:143–155

Lee CH (2000) A literature survey of wavelets in power engineering applications. Proc Natl Sci Counc ROC (A) 24(4):249–258

Li H, Yu H, Cao N, Tian H, Cheng S (2020) Applications of artificial intelligence in oil and gas development. Archives of Computational Methods in Engineering pp 1–13, https://doi.org/10.1007/s11831-020-09402-8

Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2017) Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research* 18(1):6765–6816

Lima RP, Bonar A, Coronado DD, Marfurt K, Nicholson C (2019) Deep convolutional neural networks as a geological image classification tool. Sediment Rec 17:4–9

Lin HT, Lin CJ (2003) A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. submitted to Neural Computation 3(1-32):16

Linek M, Jungmann M, Berlage T, Pechnig R, Clauser C (2007) Rock classification based on resistivity patterns in electrical borehole wall images. Journal of Geophysics and Engineering 4(2):171–183, https://doi.org/10.1088/1742-2132/4/2/006

Liu DC, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. Mathematical programming 45(1):503–528

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 2008 eighth ieee international conference on data mining, IEEE, pp 413–422

Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) 6(1):1–39

Lopez O, Berg CF, Rennan L, Digranes G, Forest T, Krisoffersen A, Bøklepp BR (2016) Quick core assessment from ct imaging: From petrophysical properties to log evaluation. In: Int. Symp. Soc. Core Anal. Snowmass Colo. USA

Malki H, Baldwin J, Kwari M (1996) Estimating permeability by use of neural networks in thinly bedded shaly gas sands. SPE Computer Applications 8(02):58–62, https://doi.org/10.2118/31010-PA

Mallat SG (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE transactions on pattern analysis and machine intelligence 11(7):674–693, https://doi.org/10.1515/9781400827268.494

Marmo R, Amodio S, Tagliaferri R, Ferreri V, Longo G (2005) Textural identification of carbonate rocks by image processing and neural network: Methodology proposal and examples. Computers & geosciences 31(5):649–659

McPhee C, Reed J, Zubizarreta I (2015) Wellsite core acquisition, handling and transportation. In: Developments in Petroleum Science, vol 64, Elsevier, pp 17–88

Mena A, Francés G, Pérez-Arlucea M, Aguiar P, Barreiro-Vázquez JD, Iglesias A, Barreiro-Lois A (2015) A novel sedimentological method based on ct-scanning: Use for tomographic characterization of the galicia interior basin. Sedimentary Geology 321:123–138, https://doi.org/10.1016/j.sedgeo.2015.03.007

Mohaghegh S, Arefi R, Ameri S, Aminiand K, Nutter R (1996) Petroleum reservoir characterization with the aid of artificial neural networks. Journal of Petroleum Science and Engineering 16(4):263–274, https://doi.org/10.1016/S0920-4105(96)00028-9

Mustra M, Delac K, Grgic M (2008) Overview of the dicom standard. In: 2008 50th International Symposium ELMAR, IEEE, vol 1, pp 39–44

Odi U, Nguyen T (2018) Geological Facies Prediction Using Computed Tomography in a Machine Learning and Deep Learning Environment. In: Proceedings of the 6th Unconventional Resources Technology Conference, American Association of Petroleum Geologists, Houston, Texas, USA, https://doi.org/10.15530/urtec-2018-2901881

O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L (2019) Keras Tuner. `https://github.com/keras-team/keras-tuner`

Pearson K (1901) Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11):559–572, https://doi.org/10.1080/14786440109462720

Perera N, Rajapakse A, Jayasinghe R (2007) On-line discrete wavelet transform in emtp environment and applications in protection relaying. In: International Conference on Power Systems Transients (IPST), pp 4–7

Porwik P, Lisowska A (2004) The haar-wavelet transform in digital image processing: its status and achievements. Machine graphics and vision 13(1/2):79–98, https://doi.org/10.1016/s0045-7906(01)00011-8

Poupon A, Hoyle W, Schmidt A (1971) Log analysis in formations with complex lithologies. Journal of Petroleum Technology 23(08):995–1005

Rafik B, Kamel B (2017) Prediction of permeability and porosity from well log data using the nonparametric regression with multivariate analysis and neural network, hassi r'mel field, algeria. Egyptian journal of petroleum 26(3):763–778, https://doi.org/10.1016/j.ejpe.2016.10.013

Rider MH (1986) The geological interpretation of well logs. Gulf Pub Co

Rogers SJ, Fang J, Karr C, Stanley D (1992) Determination of lithology from well logs using a neural network. AAPG bulletin 76(5):731–739, https://doi.org/10.1306/bdff88bc-1718-11d7-8645000102c1865d

Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:160904747

Salehi SM, Honarvar B (2014) Automatic identification of formation iithology from well log data: a machine learning approach. Journal of Petroleum Science Research 3(2):73–82, https://doi.org/10.14355/jpsr.2014.0302.04

Saljooghi BS, Hezarkhani A (2014) Comparison of wavenet and ann for predicting the porosity obtained from well log data. Journal of Petroleum Science and Engineering 123:172–182, https://doi.org/10.1016/j.petrol.2014.08.025

Schneider CA, Rasband WS, Eliceiri KW (2012) Nih image to imagej: 25 years of image analysis. *Nature methods* 9(7):671–675

Serra Ot, Abbott H (1982) The contribution of logging data to sedimentology and stratigraphy. Society of Petroleum Engineers Journal 22(01):117–131

Siddiqui S, Khamees AA (2004) Dual-energy ct-scanning applications in rock characterization. In: SPE annual technical conference and exhibition, Society of Petroleum Engineers, https://doi.org/10.2118/90520-ms

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959

Srisutthiyakorn N (2016) Deep-learning methods for predicting permeability from 2d/3d binary-segmented images. In: SEG technical program expanded abstracts 2016, Society of Exploration Geophysicists, pp 3042–3046

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958

Sudakov O, Burnaev E, Koroteev D (2019) Driving digital rock towards machine learning: Predicting permeability with gradient boosting and deep neural networks. Computers & geosciences 127:91–98, https://doi.org/10.1016/j.cageo.2019.02.002

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9, https://doi.org/10.1109/cvpr.2015.7298594

Tembely M, AlSumaiti A, et al. (2019) Deep learning for a fast and accurate prediction of complex carbonate rock permeability from 3d micro-ct images. In: Abu Dhabi International Petroleum Exhibition & Conference, Society of Petroleum Engineers, https://doi.org/10.2118/197457-MS

Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63(2):411–423, https://doi.org/10.1111/1467-9868.00293

Tschannen V, Delescluse M, Rodriguez M, Keuper J (2017) Facies classification from well logs using an inception convolutional network. arXiv preprint arXiv:170600613

Vapnik V (2000) The Nature of Statistical Learning Theory. Springer

Vapnik VN (1999) An overview of statistical learning theory. IEEE transactions on neural networks 10(5):988–999

Vapnik VN, Chervonenkis AY (1971) On the uniform convergence of relative frequencies of events to their probabilities. The Probability and its Applications 17(2):264–280

Vayatis N, Azencott R (1999) How to estimate the vapnik-chervonenkis dimension of support vector machines through simulations? Support Vector Machines: Theory and Applications", eds: Paliouras, G and Karkaletsis, V, ACAI 99

Vinegar HJ (1986) X-Ray CT and NMR Imaging of Rocks. Journal of Petroleum Technology 38(03):257–259, https://doi.org/10.2118/15277-PA, publisher: Society of Petroleum Engineers

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods 17:261–272, 10.1038/s41592-019-0686-2

Wellington SL, Vinegar HJ, et al. (1987) X-ray computerized tomography. Journal of Petroleum Technology 39(08):885–898, https://doi.org/10.2118/16983-pa

Wong P, Henderson D, Brooks L (1998) Permeability determination using neural networks in the ravva field, offshore india. SPE Reservoir Evaluation & Engineering 1(02):99–104, https://doi.org/10.2118/38034-PA

Wong PM, Gedeon TD, Taggart IJ (1995) An improved technique in porosity prediction: a neural network approach. IEEE Transactions on Geoscience and Remote Sensing 33(4):971–980, https://doi.org/10.1109/36.406683

Xie Y, Zhu C, Zhou W, Li Z, Liu X, Tu M (2018) Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances. Journal of Petroleum Science and Engineering 160:182–193, https://doi.org/10.1016/j.petrol.2017.10.028

Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights into imaging 9(4):611–629

Zhu C, Byrd RH, Lu P, Nocedal J (1997) Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS) 23(4):550–560

Part II: Scientific papers

Paper I

This paper is not included due to SPE copyright retrictions

Paper II

Research Article

Lithology classification of whole core CT scans using convolutional neural networks

Kurdistan Chawshin[1] · Carl Fredrik Berg[1] · Damiano Varagnolo[2] · Olivier Lopez[3]

Received: 3 February 2021 / Accepted: 11 May 2021

Abstract

X-ray computerized tomography (CT) images as digital representations of whole cores can provide valuable information on the composition and internal structure of cores extracted from wells. Incorporation of millimeter-scale core CT data into lithology classification workflows can result in high-resolution lithology description. In this study, we use 2D core CT scan image slices to train a convolutional neural network (CNN) whose purpose is to automatically predict the lithology of a well on the Norwegian continental shelf. The images are preprocessed prior to training, i.e., undesired artefacts are automatically flagged and removed from further analysis. The training data include expert-derived lithofacies classes obtained by manual core description. The trained classifier is used to predict lithofacies on a set of test images that are unseen by the classifier. The prediction results reveal that distinct classes are predicted with high recall (up to 92%). However, there are misclassification rates associated with similarities in gray-scale values and transport properties. To postprocess the acquired results, we identified and merged similar lithofacies classes through ad hoc analysis considering the degree of confusion from the prediction confusion matrix and aided by porosity–permeability cross-plot relationships. Based on this analysis, the lithofacies classes are merged into four rock classes. Another CNN classifier trained on the resulting rock classes generalize well, with higher pixel-wise precision when detecting thin layers and bed boundaries compared to the manual core description. Thus, the classifier provides additional and complementing information to the already existing rock type description.

Article Highlights

- A workflow for automatic lithofacies classification using whole core 2D image slices and CNN is introduced.
- The proposed classifier shows lithology-dependent accuracies.
- The prediction confusion matrix is exploited as a tool to identify lithofacies classes with similar transport properties and to automatically generate lithofacies hierarchies.

Keywords X-ray computerized tomography · Convolutional neural network · Classification · Lithofacies

---

✉ Kurdistan Chawshin, kurdistan.chawshin@ntnu.no | [1]Department of Geoscience and Petroleum, NTNU, S. P. Andersens veg 15, 7031 Trondheim, Norway. [2]Department of Engineering Cybernetics, NTNU, O. S. Bragstads Plass 2D, 7034 Trondheim, Norway. [3]Equinor ASA, Arkitekt Ebbells veg 10, 7053 Ranheim, Norway.

1 Introduction

Classifying lithofacies is an essential step toward characterizing reservoirs and better understanding their depositional environments. To predict reservoirs' saturation levels, and to perform subsequent effective reservoir modeling, it is crucial to correctly assess lithological properties such as grain size, grain shape, sorting and cementation. These lithological properties affect the petrophysical and transport properties of the reservoir rocks (e.g., porosity and permeability).

Conventional well-log interpretations performed for lithology classification typically overlook the heterogeneities below the log resolution. Currently, the whole cores extracted from wellbores are described through direct visual inspections by a team of geologists and/or petrophysicists. However, this process is time-consuming and the resulting facies classification can be affected by subjective interpretation.

The extraction of whole core data is currently requiring significant capital investment. Therefore, rapid and automated core classification and associated core analysis is seen as a key technology for enabling improved return on investments and to enhance the overall decision processes [36].

X-ray computerized tomography (CT) imaging is seen as one of the most effective nondestructive methods for inspecting whole cores at a submillimeter resolution, and the resulting digital image of the core is an aid toward the automation of the core classification process. CT images can indeed be incorporated in the classification workflow for a rapid lithology classification [10]. Whole core CT scanning has a long history in assisting the geologists to study extracted cores [39]. More precisely, 2D and 3D whole core CT scans provide high-resolution (submillimeter) information on the texture, composition and internal structure of the reservoir rocks. Moreover, whole core CT imagery may be performed in the early stages of the facies analysis process: these data can be employed before extrusion, when the core is still in an aluminum barrel [36].

From technical standpoints, each voxel in the CT images is represented by a gray-level value that indicates a certain level of X-ray attenuation. This grayscale value, and thereby the attenuation, is a function of the density and effective atomic number of the underlying material [36]. Since the first generation of the CT scanners, the scanning technique has gone through extensive refinements, and current CT images can predict 2D and 3D distribution of the chemical composition and density of the whole core [19]. This information, together with the fact that the whole core scans are stored digitally, aids laboratory analyses of the internal structure of the cores to be used in rock characterization and evaluation of plug drilling locations. Recent improvements in CT scanning and reconstruction algorithms, combined with developments in computing power and image analysis, have opened new possibilities for extracting even more information from whole cores, and thereby enhancing their value in operational settings and facilitating the automation of the core classification process.

The application of supervised and unsupervised machine learning algorithms has found significant use in many disciplines, including the petroleum industry. Recently, exploration and production companies have been extensively interested in the analysis of large data and automated solutions to reduce operational inefficiencies that slow down decision-making processes with associated losses of revenue [5].

Machine learning algorithms, especially artificial neural networks and support vector machines, have been successfully applied in several research studies to classify lithofacies and to estimate petrophysical properties using well log or core plug measurements [1, 2, 8, 10, 15, 18, 22, 25, 31, 33, 38, 41, 42, 49, 52].

In regard to image-based lithology classifications, several publications have utilized deep learning approaches to classify lithology based on the optical core photographs, borehole image logs, thin sections, and microtomographic images. De Lima et al. [12, 13] employed deep learning and transfer learning technique to classify core images of carbonate rocks. In another publication De Lima et al. [14] explored the use of deep convolutional networks to accelerate the microfacies classification based on rock thin sections. Valentin et al. [50] introduced a methodology for automatic lithofacies identification based on ultrasonic and microresistivity borehole images and a deep residual convolutional network. Baraboshkin et al. [6] compared the performance of several well-known neural network architectures (AlexNet, VGG, GoogLeNet, ResNet) to classify rock types based on the optical core images. Moreover, deep learning technique was utilized by Anjos et al. [4] to identify lithological patterns in carbonate rocks based on the microtomographic images.

In the majority of the aforementioned publications, either well log data or core analysis data have been used as inputs for the models learning phase. However, a recent trend is integrating both pieces of information together, potentially with also multiscale images. More specifically, Al-Obaidi et al. [3] used a combination of rock fabric properties extracted from image logs and well log-based petrophysical and compositional estimations to perform an automatic rock classification using a *k*-means based clustering method.

While artificial intelligence has been extensively employed for facies classification and petrophysical property estimations based on well log and core analysis data, there have been a few approaches that utilize CT images for facies classification and flow property estimations. These approaches employ information content of the CT images through the extraction of various features for clustering and classification purposes. Hall et al. [19] preprocessed the whole core CT images, extracted statistical features from processed images, and trained a Random Forest classifier to identify bioturbated core intervals. Odi and Nguyen [36] utilized physical features such as density, porosity and photoelectric effect, extracted from dual-energy CT scans, for supervised and unsupervised geological facies classifications. Moreover, the models were trained to learn the relationship between the CT extracted physical features and existing user-defined geological facies description.

Gonzalez et al. [17] considered a workflow for an automatic rock classification that combines conventional well logs, whole core CT images, optical core photographs, and routine core analysis (RCA) data. In this workflow, rock-fabric-related features are first extracted from whole core CT images and core photographs and then used to determine the rock classes by means of a clustering algorithm. Initially, the authors assumed several rock classes, and then they optimized this number by iteratively increasing the number of classes and minimizing a permeability-based cost function below a certain threshold. The obtained rock classes were finally used to train an artificial neural network to predict the classes from well log data. shin et al. [10] employed Support Vector Machines (SVM) to automatically classify lithofacies using the first order statistics and gray-level co-occurrence matrix (GLCM) features extracted from 2D cross-sectional whole core CT images. The authors used an SVM model to learn the relationship between the extracted features and expert-derived manual core descriptions.

In the mentioned publications, facies classification is performed using information content of the CT images in the form of various statistical and textural features. However, the CT images are not directly used as input for machine learning-based classifications.

In this study, we propose a workflow for automatic lithofacies classification that uses whole core CT image slices as input to train a CNN model. In the proposed approach, the need for manual feature extraction is eliminated as relevant features are learned by the network while it is being trained on a set of CT images. The obtained results reveal that the trained classifier is able to distinguish certain lithofacies classes with satisfying accuracy. However, lithofacies classes with similar texture and grayscale values are confused. In our workflow, the information acquired from prediction results is utilized to evaluate the misclassified lithofacies classes in terms of similarities in the transport properties. Further, as a post-classification processing step, hierarchical clustering analysis is performed to automatically cluster similar lithofacies classes using the prediction confusion matrix and then these results, together with porosity–permeability relationships, are used to group 20 lithofacies classes into 4 rock classes.

2 Methodology

In brief, we propose an automatic lithofacies classification workflow that uses whole core CT images and CNN and that is summarized in Fig. 1. The whole approach starts with preprocessing of 2D DICOM (Digital Imaging and Communication in Medicine) images. Lithofacies labels are then assigned to the processed images based on a user-defined geological core description. Lithofacies simply refers to a lithological subdivision that is distinguishable by its texture, grain size and the depositional environment. The labeled images are further augmented and used as inputs to train a CNN classifier. The trained classifier is then validated on a set of unseen images to predict lithofacies classes. Then, lithofacies classes that are deemed to be sufficiently similar are combined into rock classes (i.e., a combination of similar lithofacies classes form a rock class); in this step, the similarity indexes are computed starting from assessments of the transport properties (porosity and permeability) together with the degree of confusion in the confusion matrix resulting from the learning algorithm. Further, the classifier is coarsened with respect to

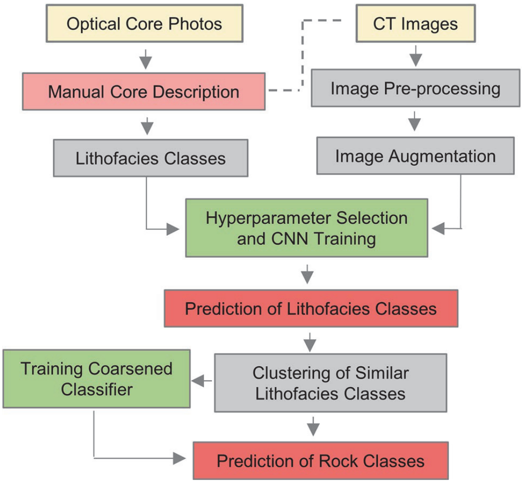


Fig. 1 Proposed workflow for lithofacies classification

the number of rock classes; in other words, the classifier is trained with a smaller number of rock classes. Finally, the coarsened classifier is employed to predict rock classes on a set of unseen images.

In the following subsections, the CNN algorithm and its general architecture will be explained in detail, followed by image preprocessing and image augmentation processes employed in this study.

2.1 Convolutional neural networks

Convolutional neural networks (CNN) have found significant applications in many sciences and industries. They have proven to be specifically effective in the fields of image recognition, voice recognition and classification. In general, neural networks draw the inspiration from the human brain. As mentioned before, this class of algorithms learns the relevant features directly from the input training data, so there is no need for manual feature extraction by a subject matter expert. Most of the modern CNN architectures consist of alternating convolutional and pooling layers followed by fully connected layers. The convolutional and pooling layers deal with feature extraction, while the fully connected layers map these extracted features into the final output. For an extensive discussion on CNN, we refer the interested reader to [53].

In the convolutional layers, a convolution operation is performed, i.e., a set of optimizable convolutional kernels are superposed in each position of the image represented by a 2D array of pixels. An element-wise multiplication between the elements of the kernel and the receptive field in the input image is performed, and the product results are summed up and stored in the corresponding position in the output feature map. Once the convolution operation is computed and stored for that specific location, the kernel is then moved either horizontally or vertically by an offset called *stride*. This process is repeated until the entire image is covered and the resulting feature map is completely populated. Convolutional layers are locally connected, whereas in the classic neural networks each neuron is fully connected to the neurons in the other layers.

To introduce nonlinearity, the outputs of the convolution operations pass through an activation function. The most common activation function is the rectified linear unit (ReLU); the advantage of using this specific function is that it allows fast and effective convergence during the training process. The feature map output of the convolutional layer records the exact position of the existing features in the input image. Therefore, minor spatial changes in the input image will yield a different feature map. To address this problem, a pooling layer is added after applying the nonlinear activation function (e.g., ReLU) to the feature map output of the convolution operation. A pooling operation is selected to be applied on each individual feature map. Two common pooling functions are average pooling and maximum pooling. The advantage of the added pooling layer is that the pooled feature map becomes invariant to local translations and spatial variations in the input image, e.g., edges, angles, feature positions, etc. [24].

The downsampled feature map outputs derived from the final pooling layer are then flattened into a 1D array of values that is connected to one or more fully connected layers that are referred to as dense layers. Here, input nodes are connected to output nodes by learnable weights [53]. The extracted features are eventually mapped into the final output of the network through the fully connected layers. Nonlinearities may also be introduced in the fully connected layers by adding an activation function (such as ReLU) following each fully connected layer.

Note that the activation function applied to the final fully connected layer is normally different than the other layers, and it is selected depending on the type of the task, i.e., classification and regression. A common activation function for multiclass classification is the so-called ”softmax” function that returns the probability distribution of the predicted classes, i.e., it converts the output of the last layer into the predicted output class probabilities.

2.2 Information on the type of available data

The provided CT scan data consist of individual cross-sectional image slices from each core interval. Therefore, the number of image slices differ for each core, since depending on both the length of the core itself and the corresponding vertical image resolution (i.e., how many images are taken per meter of core). As an example, if the vertical image resolution is 0.4 millimeters and an individual core length is 1 meter, this results in more than 2000 individual image files for that 1 meter core interval. In our dataset, the image slices are stored in a 16-bit unsigned DICOM format, a standard format developed for medical images [34]. The DICOM images of individual cores have been then stacked together and stored as 3D raw images using the ImageJ software [43].

2.3 Image preprocessing

To prepare the images as inputs for our CNN training process, we need to discard undesired noncore regions. The images coming from certain zones can negatively affect the classification results, since they contain information that is nonrelated to the actual phenomena we want to model.

The first step we adopted is to remove border effects by cropping the 3D raw image slices into rectangular crops of size 256 × 256 pixels. A comparison of an example image before and after this cropping is shown in Fig. 2. After cropping, a global minimum and maximum intensity value, selected by observing the 3D histograms of all rectangular crops, is assigned to the images of the entire considered core intervals. Further, the intensity adjusted images are encoded in 8-bit format, i.e., 0– 255 gray-scale, and stored for further analysis.

Another preprocessing operation includes removing images with missing data associated with poor core recovery, induced fractures, or rush plugs. Note that the image slices with missing core intervals show low gray-level attenuation values (Fig. 3a).

We also note that the images dataset contains a number of other undesired artefacts related to core barrel couplings, drilling mud invasion, and cementation of high-density minerals such as pyrite and siderite (examples are shown in Fig. 3b and c). Also these zones need to be excluded from the training set.

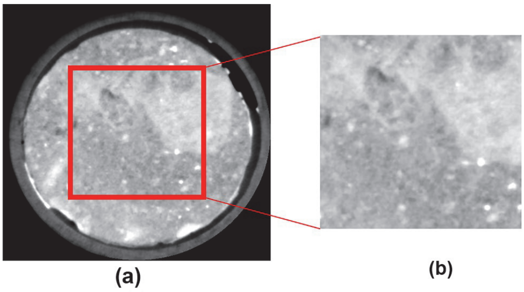

(a) (b)

Fig. 2 Original DICOM image slices **a** are cropped (red square) into 256x256 squares **b** as a preprocessing step to prepare images to be used as inputs for the CNN training process

To flag and remove the above artefacts, we thus implemented dedicated type-dependent algorithms to the raw data. More precisely, to remove missing intervals we calculate the average attenuation μ_c in the center of the image using a centered square covering 40% of the total number of pixels. If the computed average attenuation is less than a predefined cutoff C_m, the image is flagged and removed:

$$f_m = \begin{cases} 1, \text{ if } \mu_c < C_m \\ 0, \text{ else} \end{cases} \tag{1}$$

where f_m is the flag for missing interval. The image is removed if f_m is equal to 1.

Intervals with high-density material appear very bright with relatively high gray-level attenuation readings. To identify these intervals, the average attenuation μ of the whole 2D image is computed and, if the average is greater than a predefined cutoff C_h, the image is flagged for removal:

$$f_h = \begin{cases} 1, \text{ if } \mu > C_h \\ 0, \text{ else} \end{cases} \tag{2}$$

where f_h is the high-density flag. The image is removed if f_h is equal to 1.

In the intervals with core barrel couplings, the attenuation values in the middle of the images are lower than the attenuation values of the image edges (i.e., the edges are brighter, as shown in Fig. 3c). To detect intervals with core barrel couplings, the difference in average attenuation of the center and edges of the 2D image is calculated. As above, the center average attenuation μ_c is computed considering 40% of the total number of pixels using a centered square. To represent edge average attenuation μ_e, the outer 5% of the total number of pixels along the edges are considered. If the difference between center average attenuation and edge average

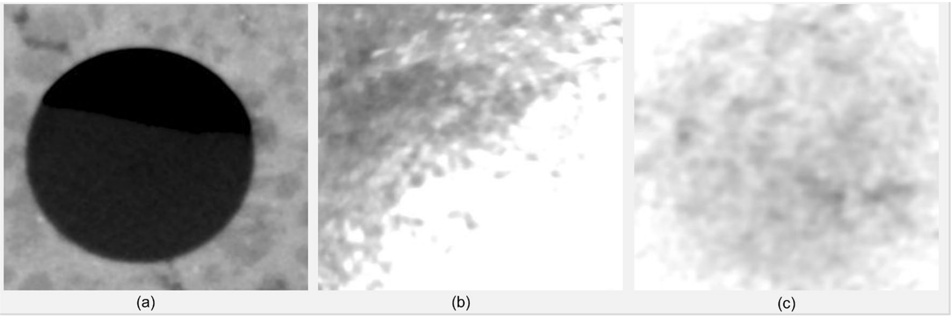


Fig. 3 2D image slices with: **a** missing CT values (due to rush plugs), **b** high-density material (cementation or drilling mud invasion), and **c** core barrel coupling

attenuation is greater than a predefined cutoff value C_b, the image interval is flagged and removed:

$$f_b = \begin{cases} 1, \text{ if } \mu_e - \mu_c > C_b \\ 0, \text{ else} \end{cases} \tag{3}$$

where f_b is the core barrel coupling flag. The image is removed if f_b is equal to 1. Note that the thresholds above have been computed using the global distribution of the minimum, mean and maximum intensity values observed in the dataset.

Finally, to reduce computational time associated with CNN training, the remaining 2D CT images are coarsened by a factor of four (i.e., the final image size is 64 × 64 pixels). Further, the images are rescaled, i.e., all pixel values are divided by 255, before being used as input for the CNN training.

2.4 Image augmentation

Generally, large amounts of training data are required to achieve a good performance in deep neural networks. Image augmentation is a strategy that is performed to boost the performance of the network through different kinds of modifications, e.g., random rotation, shifting, shearing and flipping, applied to the original images.

Image augmentation is applied during the training phase, so that the model can learn from more image examples. In our framework, we specifically considered rotation and horizontal flips of the original images. We thus implemented the ”ImageDataGenerator” class in Python using the Keras API [11], a publicly available code that can be used for image augmentation purposes on the fly. The ”ImageDataGenerator” class rotates the images randomly within a range of user-defined angles. Therefore, in case of squared images, it is very likely that for some specific rotation angles, the pixels will fall out of the image frame leaving some areas of the image with no pixels. There are a number of interpolation techniques such as nearest neighbor that can be used for those areas, but it can amend the key features resulting in dissimilar features counterproductive for training. To avoid this problem, the images were rotated outside Keras, while the horizontal flip was applied in Keras using ”ImageDataGenerator” class on the fly during training the CNN classifier. The images were rotated by 90°, 180° and 270°. An example of the rotated and horizontally flipped images is shown in Figs. 4 and 5.

3 The dataset

3.1 Whole core CT scan images

This study uses whole core cross-sectional image slices from a well on the Norwegian continental shelf. The

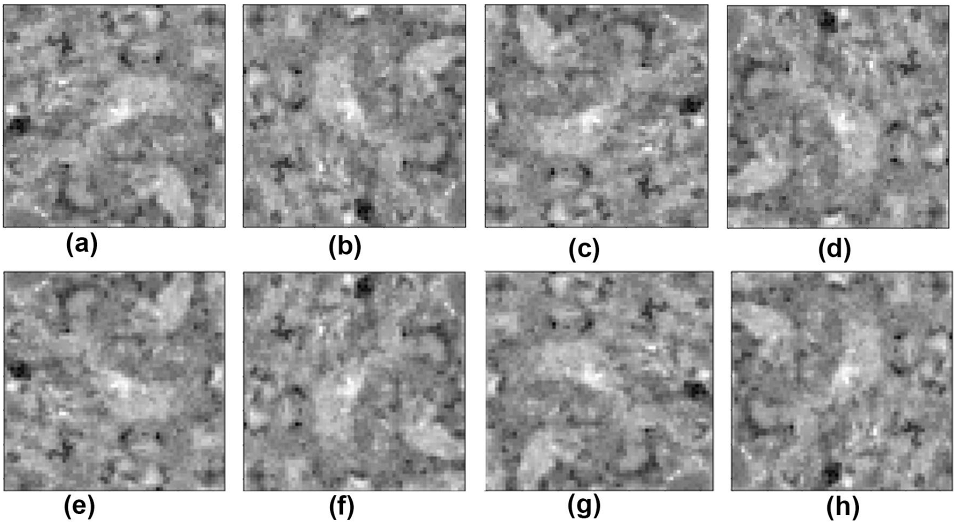


Fig. 4 An example of image augmentation applied on the CT images. **a** Original CT image, **b** original image rotated by 90 degrees, (**c**) original image rotated by 180 degrees, **d** original image rotated by 270 degrees, **e** original image horizontally flipped, **f** 90° rotated and horizontally flipped, **g** 180° rotated and horizontally flipped, **h** 270° rotated and horizontally flipped. Note that the images are coarsened by a factor of 4 with a final size of 64 × 64 pixels

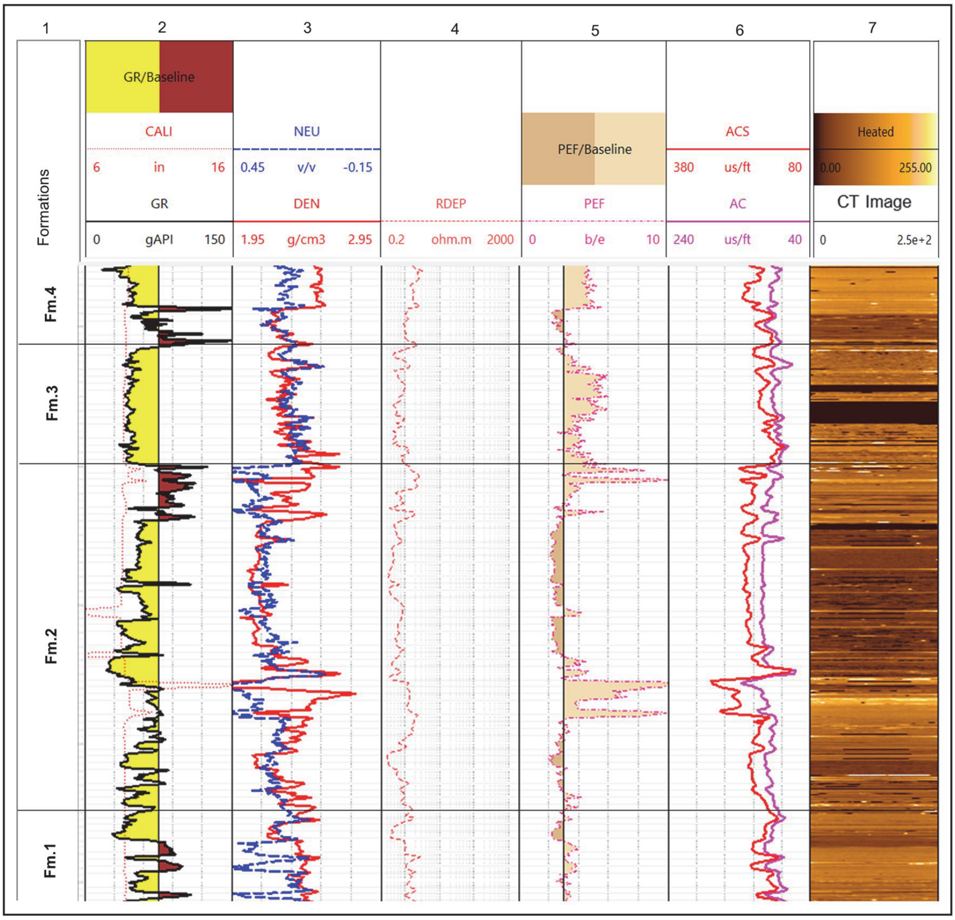


Fig. 5 Well log data and 2D cross section of the core CT image showing 142 meters of the studied well. Log tracks from left to right: track 1: Formations, track 2: Caliper (CALI) and Gamma ray (GR), track 3: Density (DEN) and Neutron (NEU), track 4: Deep resistivity (RDEP), track 5: Photoelectric factor (PEF), track 6: Compressional wave slowness (AC) and shear wave slowness (ACS), track 7: 2D cross section of whole core CT scan

studied well penetrates four main formations denoted as Fm.1, Fm.2, Fm.3 and Fm.4 in Fig. 5: Formation 1 consists of very fine-grained argillaceous sandstones and cemented sandstones, Formation 2 constitutes successive layers of mudstones and fine-grained sandstones, Formation 3 consists of granule-rich medium-grained sandstones and spiculites (a biogenic rock composed of sponge silica spicules), and Formation 4 comprises mud and calcite rich marlstones. As mentioned in Sect. 2.3, the images were provided in 16-bit unsigned DICOM slices with a vertical resolution of approximately 0.45 millimeters. The individual DICOM images were stacked and stored as 3D raw images and then cut into rectangular crops. In addition, a global minimum and maximum intensity value was assigned to all images before they were encoded in 8-bit format. The images with undesired artefacts were removed as described above, and the remaining images were coarsened by a factor of four to reduce computational time.

3.2 The lithofacies from the employed core description

We exploit information obtained from a manual core-based lithology description, performed by a geologist, as groundtruth to create the training lithofacies classes. The CNN classifier was then trained to learn the relationship between the image features extracted by the convolution process, and the corresponding lithofacies classes. For the

Table 1 Lithofacies classes and their associated fractions derived from core-based lithology descriptions (225524 images from 142 meter of core)

Lithofacies labels	Description	Fraction
Marl	Mud/clay rich marl	0.0214
CalMarl	Marl with caliche cementation	0.0157
SpiculiteSS	Medium-grained spiculitic sandstone	0.0438
Mudstone	Dark gray mudstone with plain parallel bedding, mottled mudstone	0.1181
WCemBelSS	Well-cemented medium-grained sandstone with Belemnite fossils	0.0035
GraMSSDispC	Granule-rich medium-grained sandstone with dispersed carbonate cementation	0.103
PCemGraMSS	Poorly cemented granule-rich medium-grained sandstone	0.032
WCemMSS	Well-cemented medium-grained sandstone	0.025
MudsHighDens	Mudstone with high density minerals (pyrite)	0.005
ArgFineSS	Argillaceous fine-grained sandstone	0.0726
RippleFineSS	Fine-grained sandstone with ripple cross-lamination	0.0809
MassFineSS	Massive fine-grained sandstone	0.099
CrossFineSS	Fine-grained sandstone with cross-stratified lamination	0.093
MudFineSS	Muddy fine-grained sandstone	0.0397
BioFineSS	Bioturbated fine-grained sandstone	0.0121
WCemFineSS	Well-cemented fine-grained sandstone	0.013
ContMud	Continental mudstone	0.0906
MassVeryFineSS	Massive very fine-grained green sandstone	0.0385
CemVeryFineSS	Cemented very fine-grained green sandstone	0.0667
VeryFineSSHorizontal	Very fine-grained sandstone with horizontal lamination	0.0264

sake of completeness, we report that our dataset presents 20 lithofacies classes derived through the manual core description mentioned above (the abbreviated classes together with a short description is found in Table 1). The three most abundant lithofacies are mudstone (marine and continental), granule-rich medium-grained sandstone with dispersed cementation, and fine-grained sandstone with different textures/laminations (ripple, cross-stratified and massive); these are interbedded with other sparser lithofacies.

4 Training phase

In this section, the training phase will be explained in detail. The section starts with the strategy used to separate train and test samples followed by training steps and hyperparameter optimization processes.

4.1 Division of the dataset in training vs. test data

A standard data analysis paradigm is to train a machine learning model on a set of data considered as the groundtruth and then evaluate its statistical performance on another set of unseen instances, again considered as correctly labeled in the manual labeling process. Considering the statistical distribution of the images in our dataset, we assessed that a suitable training vs. test sets splitting ratio is 80% for training and 20% for testing. To maintain continuous intervals and at the same time balancing the frequency of the lithofacies within each set, the train and test sets were selected manually. The reason for not selecting train and test sets randomly is that the images are slowly varying, so a random selection would give similar data points in both sets. Approximately 20% of the train set was employed as validation set, which is used to evaluate the performance of the model during training (see Sect. 4.2.3).

For completeness, the distribution of different lithofacies classes in the resulting train and test sets is presented in Fig. 6, from which we can see similar class distributions in both sets.

4.2 Details on the CNN training process

The CNN training is a process by which the kernels weights in the convolutional layers, the weights in the fully connected layers, and their associated biases are adjusted in such a way that the difference between the predicted labels and the given labels (i.e., the groundtruth) is minimized. Training is commonly performed by a forward- and back-propagation process throughout the entire network using a gradient descent optimization algorithm and a loss function. The loss function computes the difference between the output predictions, computed through forward propagation, and the actual label. The network

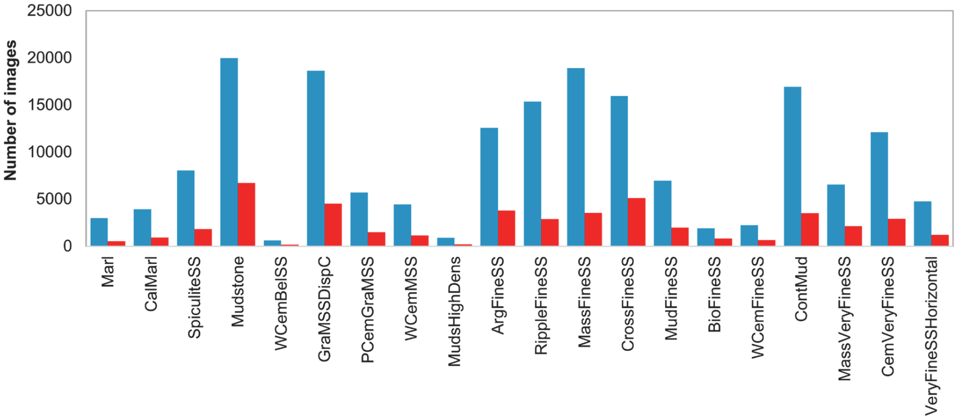


Fig. 6 Distribution of different lithofacies classes in the train (blue) and test (red) sets

performance is evaluated using the loss function. Cross-entropy is typically used as the loss function for multiclass classification tasks, whereas the mean squared error is typically used for prediction of continuous values, i.e., regression analysis [53]. In the current study, we are dealing with a multiclass classification task. Therefore, we used cross-entropy to determine the loss function of the CNN model as given by [9]:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(p\left(\hat{y}_i\right)\right), \tag{4}$$

where y_i and $\hat{y}_i$ are, respectively, the true and predicted labels of the i^{th} sample, p is the probability, while N is the total number of training samples.

As mentioned above, the learnable parameters are updated iteratively using a gradient descent optimization algorithm that seeks to minimize cross-entropy losses. Basically, the partial derivative of the loss function with respect to each learnable parameter is first calculated; once the whole loss function gradient is computed, the learnable parameters are updated using [53]:

$$w^{+} = w - \alpha\frac{dL}{dw} \tag{5}$$

where w refers to each learnable parameter with w^{+} being the updated value, α stands for learning rate, and L is the loss function. The learning rate is an important hyperparameter that determines how fast the learnable parameter (e.g., weight) should move in the direction of the gradient. Note that finding the optimal learning rate during training is crucial for neural networks, since the training process may not converge when using a too high learning rate (in this case, indeed, the optimizer overshoots the minimum and lands in a zone of the parameters space that leads to worse loss values).

To avoid this issue, it is common to employ various types of optimizers so to search the optimum weight and kernel parameters using a pool of different gradient descents strategies, among which then choose the best one. Examples of the different types of descent methods are stochastic, batch and mini-batch gradient descents. These methods vary in terms of the number of samples used to compute the error between the actual and predicted labels.

In our study, we evaluated the performance of the RMSProp [20] and Adam [29] optimizers to optimize the weights. The obtained results revealed that Adam outperformed the RMSProp. Therefore, we eventually optimized the weights using the Adam optimizer together with a mini-batch gradient descent method. Note that this is the most common variation of gradient descent used in deep learning; to give some intuitions, mini-batch gradient descent splits the training data into small batches and calculates the error per batch before updating the learnable parameters.

In our study, the final optimal approach was to consider a batch size of 32 images and a CNN classifier training process of 70 epochs (where an epoch is a period in which all the training samples have been presented at least once to the network).

4.2.1 Hyperparameter selection

Generally, there exist two types of parameters in the machine learning algorithms. As mentioned in Section 4.2, the kernel weights in the convolutional layers, the weights in the fully connected layers, and their associated biases are learnable, and thus optimized during the training process. The second type of parameters, referred to as hyperparameters, determine the structure of the cost function that is minimized, and need to be set by the user. These hyperparameters include the learning rate, the number of convolutional layers, the number of kernels in the convolutional layers, and the number of neurons in the fully connected layers. It is quite straightforward to realize that the performance of a machine learning model is highly dependent on the right choice of both the parameters and the hyperparameters. The process of adjusting the hyperparameters is called hyperparameter tuning.

As previously explained, the here proposed CNN classifier was developed in Keras using the Tensorflow backend. In our case we solve the hyperparameter tuning problem using the Keras tuner library [30, 37, 40]. This library enables to define a search space that includes the considered hyperparameters and an opportune tuner that will automate the solution of this tuning process. More precisely, the task of the tuner is to evaluate a certain number of hyperparameter combinations in a model that is explicitly set-up for hypertuning, i.e., a *hypermodel*. The considered hyperparameters in this study are presented in Table 2. Four tuners are available is Keras, including `RandomSearch`, `Hyperband`, `BayesianOptimization`, and `Sklearn`. For more information on the differences among these approaches, we direct the interested reader to [7, 23, 32, 47].

In this study, we utilize the Hyperband algorithm [32], a relatively new method for tuning the iterative algorithms. Basically, the strategy behind this approach is to try a large number of random configurations using adaptive resource allocation and an early stopping rule to quickly converge to a high-performance model. More specifically, the random configurations are run for a specific number of epochs (i.e., one or two) per configuration, and then the top-performing model configurations based on the previous results are trained for longer runs. Finally, the algorithm returns a best configuration trained to the assigned maximum number of epochs. The optimized classifier architecture, obtained by this hyperparameter selection processes, is presented in Fig. 7, and described in detail in the next section.

4.2.2 Classifier architecture

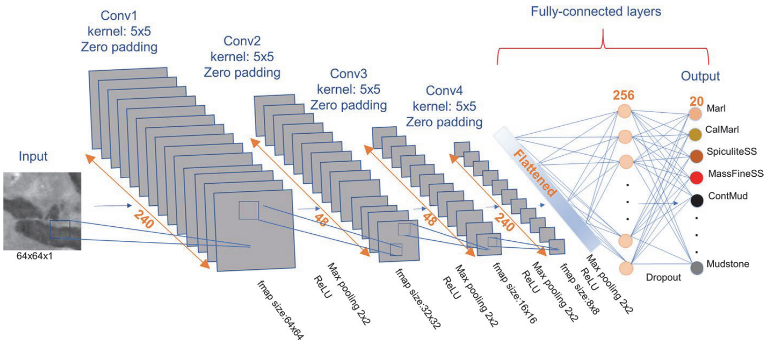
The proposed CNN classifier architecture is shown in Fig. 7. Its input and output layers consist of 2D image slices and lithofacies classes that have been derived from the available core descriptions. The classifier employs four distinct convolutional layers, indicated as ″`Conv1`,″ ″`Conv2`,″ ″`Conv3`,″ and ″`Conv4`,″ with 240, 48, 48 and 240

Table 2 Potential hyperparameters and the potential search space used in this work during the hyperparameter selection

Training hyperparameters	Parameter space
Number of convolutional layers	(1, 2, 3, **4**)
Number of convolutional kernels (filters)	(16, **48**, 80, 112, 144, 176, 208, **240**)
Kernel size	(3, **5**)
Learning rate	(0.01, 0.001, **0.0001**)
Number of neurons in the fully connected layer	(32, 64, 96, 128, 160, 192, 224, **256**)
Dropout rate	(0, **0**.2, 0.4, 0.6)

The final optimal values are shown in bold. Note that two numbers are bold for convolutional kernels since two convolutional layers have 48 kernels each, while the other two have 240 kernels each (Fig. 7)

Fig. 7 Proposed CNN architecture for lithofacies classification



convolutional kernels, respectively. Note that here we employ a kernel size of 5 × 5; this specific dimension was indeed resulting as optimal from the hyperparameter tuning process, and has been used in all our convolutional layers.

In order to preserve the original image size, we moreover applied a zero padding technique in each convolutional layer; i.e., we added a layer of pixels with values of zero around the image edges. The convolution operation in each layer is in our scheme then performed using a stride of 1, and the resulted feature maps are passed through a ReLU activation function to introduce nonlinearity. In our context the stride is, basically, the number of pixel shifts when the kernels are moved throughout the input image. After applying the ReLU function, the feature maps are sent to the subsequent pooling layer, where they are downsampled using a max pooling layer with pooling window size of 2 × 2 and a stride of 1.

The pooled feature maps of the last convolutional layer are flattened into a one-dimensional vector that is connected to the output layer in the fully connected layer. The proposed network contains one hidden layer with 256 neurons. As mentioned before, the number of neurons in the hidden layer is a hyperparameter that was optimized during hyperparameter tuning. A ReLU function is also applied to the hidden layer followed by the dropout layer. Dropout is a regularization technique, where randomly selected neurons are discarded during training (i.e., they are temporarily removed from the network together with their incoming and outgoing connections). The dropped-out neurons are not employed in the backpropagation phase [21, 48]. A dropout rate of 0.2 was applied in the proposed network meaning that one in 5 of the neurons in the hidden layer will be randomly ignored from each update iteration. As mentioned in Table 2, the dropout rate is a hyperparameter that, as the others, is optimized during the hyperparameter tuning phase. This regularization scheme is meant to prevent overfitting, and can be interpreted as an attempt to optimize the bias-variance tradeoff of the overall estimator. For more details about the statistical interpretations of regularization see [44].

Another common regularization technique in deep learning is *batch normalization*. In batch normalization, the output of a convolutional layer is normalized before being used in the next one. This technique is known to have also a regularization effect, and it is empirically known to typically speedup the network training, plus make it less sensitive to the initialization point [26]. We note that, however, this is not guaranteed in general settings—and indeed, in the current study, more accurate results were obtained without using batch normalization. The last layer in Fig. 7 is the output layer with 20 nodes corresponding to the 20 lithofacies labels. The proposed architecture provides 1’628’612 trainable parameters.

4.2.3 Classifier evaluation

As mentioned previously, 20% of the training images were utilized as the validation set. The cross-entropy loss and accuracy were considered as training metrics to evaluate the performance of the CNN classifier during training. Figure 8 shows how the accuracy and cross-entropy change over time during the training process. As one can see from the plots, the classification accuracy increases with increasing number of epochs in both the training and validation sets. However, the cross-entropy loss decreases with increasing number of epochs. The training metrics start to converge at around 70 epochs.

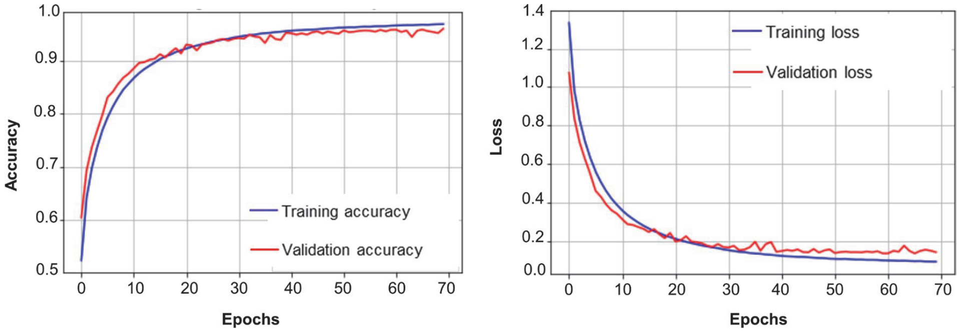


Fig. 8 Model performance on the training and validation set. The plot to the left shows the accuracy results by increasing the number of epochs, whereas the plot to the right shows loss results by increasing the number of epochs

5 Results

The lithofacies classification results acquired by using the trained classifier on a set of unseen images will be presented and discussed in the following sections.

5.1 Lithofacies prediction

To evaluate the performance of the trained CNN classifier on unseen data, the model was used to predict lithofacies in another part of the well, previously denoted as the test set. For consistency, the test images are passed through the same processes of image preprocessing and rescaling before being actually classified. The corresponding prediction accuracy metrics and confusion matrix calculated by cross-classifying the lithofacies classes from core description (classification groundtruth) and CNN prediction are summarized in Table 3 and Fig. 9. Here, accuracy is defined as the sum of true positives divided by total number of samples in the test set (i.e., probability of correct classification). Precision is quantified as the sum of true positives divided by the sum of true positives and false positives across all the lithofacies classes in the test set. In other words, precision represents the probability that the predicted lithofacies class, given the classification results for individual images, actually belongs to that class. Recall is calculated as the sum of the true positives divided by the sum of true positives and false negatives across all the lithofacies classes. Precision and recall results are combined into a single measurement, i.e., the f1-score, through the following formula:

$$\text{f1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{6}$$

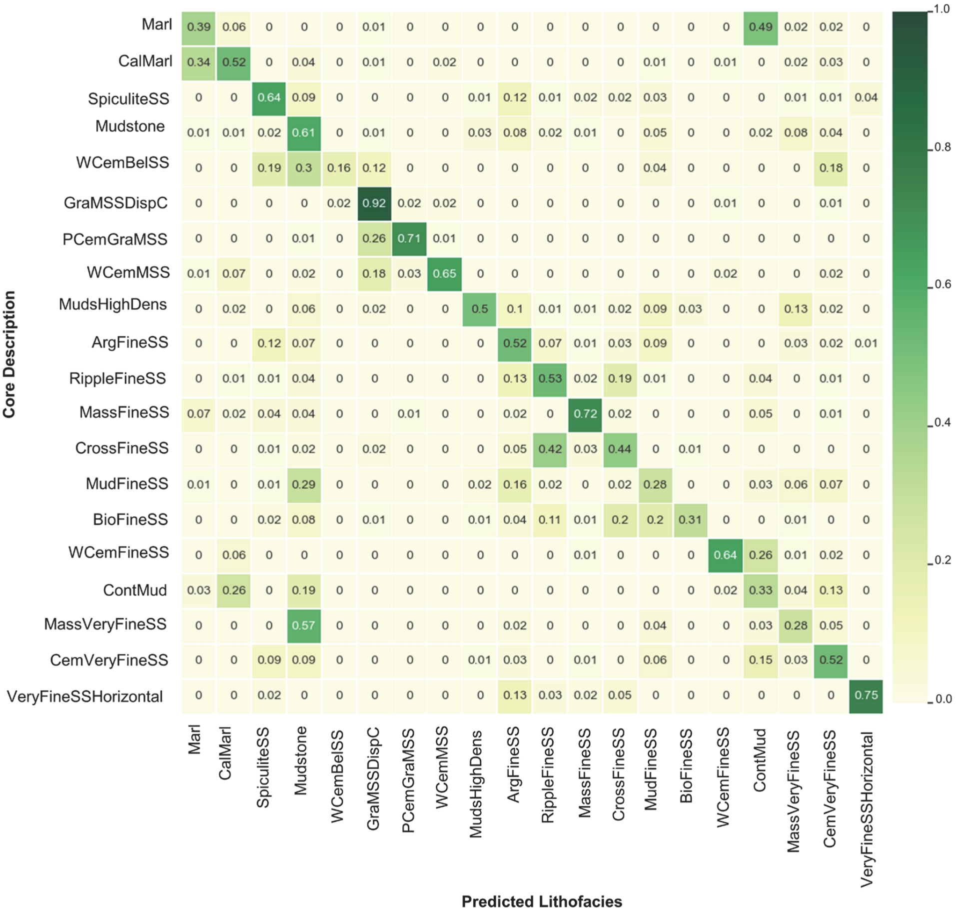
The confusion matrix provides information on the similarity of the lithofacies classes in the confusion space. If we consider each row in the confusion matrix as a vector representing a particular class, the i^{th} coordinate in that row vector shows the degree of misclassification of the considered class with the i^{th} class [16]. In other words, the diagonal values of the confusion matrix represent the recall, while the off-diagonal values correspond to the degree of misclassifications. All the row vectors in the computed confusion matrix are normalized to one. Therefore, in the case of 100% accuracy, the i^{th} coordinate of the i^{th} row vector will be 1, while all the off-diagonal coordinates will be 0.

Looking at the confusion matrix in Fig. 9, we observe that the proposed classifier is able to predict some of the lithofacies classes with recall values above 0.7. More specifically, granule-rich medium-grained sandstones with dispersed calcite cementation record the highest recall (0.92), followed by very fine-grained sandstones with horizontal lamination (0.75), massive fine-grained sandstones (0.72) and poorly cemented granule-rich medium-grained sandstones (0.71). However, the classifier misclassifies the other lithofacies into another class or a set of classes with different degrees of confusion.

In particular, the classifier misclassifies very fine-grained lithofacies classes, i.e., marl, marl with caliche cementation, mudstone, mudstone with high density minerals, muddy fine-grained sandstone, cemented very fine-grained green sandstone, massive very fine-grained green sandstone, and continental mudstone. Examples of these misclassified lithofacies classes are illustrated in Fig. 10, from which we can see that these lithofacies classes actually show similar texture and grain sizes, therefore similar gray-scale values, with no distinct features. This explains the difficulties that the classifier encounters in doing its designed task. As lithofacies with similar grayscale and textural properties are expected to exhibit similar transport properties, porosity and permeability data from core analysis measurements were used to investigate the transport properties of the classified lithofacies. Figure 11 shows the porosity–permeability cross-plot for core plug samples from the same core data as in our CT images, where different colors correspond to the different lithofacies that have

Table 3 Prediction accuracy metrics on the test set using the trained CNN classifier. Support shows the number of predicted samples for each class

Lithofacies labels	Precision	Recall	F1-score	Support
Marl	0.23	0.39	0.29	542
CalMarl	0.27	0.52	0.36	918
SpiculiteSS	0.50	0.64	0.56	1835
Mudstone	0.53	0.61	0.56	6684
WCemBelSS	0.19	0.16	0.17	160
GraMSSDispC	0.83	0.92	0.87	4498
PCemGraMSS	0.84	0.71	0.77	1491
WCemMSS	0.82	0.65	0.72	1161
MudsHighDens	0.26	0.50	0.34	187
ArgFineSS	0.48	0.52	0.50	3774
RippleFineSS	0.36	0.53	0.43	2879
MassFineSS	0.86	0.72	0.78	3522
CrossFineSS	0.68	0.44	0.53	5096
MudFineSS	0.30	0.28	0.29	1979
BioFineSS	0.79	0.31	0.44	824
WCemFineSS	0.72	0.64	0.68	653
ContMud	0.44	0.33	0.38	3489
MassVeryFineSS	0.36	0.28	0.32	2121
CemVeryFineSS	0.54	0.52	0.53	2906
VeryFineSSHorizontal	0.86	0.75	0.80	1192



	Marl	CalMarl	SpiculiteSS	Mudstone	WCemBelSS	GraMSSDispC	PCemGraMSS	WCemMSS	MudsHighDens	ArgFineSS	RippleFineSS	MassFineSS	CrossFineSS	MudFineSS	BioFineSS	WCemFineSS	ContMud	MassVeryFineSS	CemVeryFineSS	VeryFineSSHorizontal
Marl	0.39	0.06	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0.49	0.02	0.02	0
CalMarl	0.34	0.52	0	0.04	0	0.01	0	0.02	0	0	0	0	0	0.01	0	0.01	0	0.02	0.03	0
SpiculiteSS	0	0	0.64	0.09	0	0	0	0	0.01	0.12	0.01	0.02	0.02	0.03	0	0	0	0.01	0.01	0.04
Mudstone	0.01	0.01	0.02	0.61	0	0.01	0	0	0.03	0.08	0.02	0.01	0	0.05	0	0	0.02	0.08	0.04	0
WCemBelSS	0	0	0.19	0.3	0.16	0.12	0	0	0	0	0	0	0	0.04	0	0	0	0	0.18	0
GraMSSDispC	0	0	0	0	0.02	0.92	0.02	0.02	0	0	0	0	0	0	0	0.01	0	0	0.01	0
PCemGraMSS	0	0	0	0.01	0	0.26	0.71	0.01	0	0	0	0	0	0	0	0	0	0	0	0
WCemMSS	0.01	0.07	0	0.02	0	0.18	0.03	0.65	0	0	0	0	0	0	0	0.02	0	0	0.02	0
MudsHighDens	0	0.02	0	0.06	0	0.02	0	0	0.5	0.1	0.01	0.01	0.02	0.09	0.03	0	0	0.13	0.02	0
ArgFineSS	0	0	0.12	0.07	0	0	0	0	0	0.52	0.07	0.01	0.03	0.09	0	0	0	0.03	0.02	0.01
RippleFineSS	0	0.01	0.01	0.04	0	0	0	0	0	0.13	0.53	0.02	0.19	0.01	0	0	0.04	0	0.01	0
MassFineSS	0.07	0.02	0.04	0.04	0	0	0.01	0	0	0.02	0	0.72	0.02	0	0	0	0.05	0	0.01	0
CrossFineSS	0	0	0.01	0.02	0	0.02	0	0	0	0.05	0.42	0.03	0.44	0	0.01	0	0	0	0	0
MudFineSS	0.01	0	0.01	0.29	0	0	0	0	0.02	0.16	0.02	0	0.02	0.28	0	0	0.03	0.06	0.07	0
BioFineSS	0	0	0.02	0.08	0	0.01	0	0	0.01	0.04	0.11	0.01	0.2	0.2	0.31	0	0	0.01	0	0
WCemFineSS	0	0.06	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0.64	0.26	0.01	0.02	0
ContMud	0.03	0.26	0	0.19	0	0	0	0	0	0	0	0	0	0	0	0.02	0.33	0.04	0.13	0
MassVeryFineSS	0	0	0	0.57	0	0	0	0	0	0.02	0	0	0	0.04	0	0	0.03	0.28	0.05	0
CemVeryFineSS	0	0	0.09	0.09	0	0	0	0	0.01	0.03	0	0.01	0	0.06	0	0	0.15	0.03	0.52	0
VeryFineSSHorizontal	0	0	0.02	0	0	0	0	0	0	0.13	0.03	0.02	0.05	0	0	0	0	0	0	0.75

Fig. 9 Confusion matrix for the test set prediction

been derived from the manual core description. Here, we can see that the aforementioned misclassified lithofacies fall into the same region with porosity and permeability values less than 0.20 and 10 mD, respectively (marked by red ellipsoid in Fig. 11).

Likewise, fine-grained sandstones with horizontal lamination, ripple cross-lamination and cross-stratified lamination are not classified with satisfying performance. The porosity–permeability cross-plot shows that these lithofacies, together with massive fine-grained sandstones, fall in the same region in the porosity–permeability cross-plot. More specifically, they exhibit porosity values above 0.28 and permeability values ranging from 100 mD to approximately 30 Darcy (represented by the blue ellipsoid).

Granule-rich medium-grained sandstone samples (P-CemGraMSS, GraMSSDispC) spread out in the regions with permeability values ranging from 30 mD up to 50 Darcy (the green ellipsoid). However, most of the samples belonging to these classes exhibit porosity and permeability values above 0.20 and 1 Darcy, respectively. The prediction results shown in Fig. 9 indicate that poorly cemented granule-rich sandstone (PCemGraMSS) lithofacies are mainly misclassified as granule-rich sandstone with dispersed calcite cementation. The spiculite sandstone samples exhibit porosity values ranging from 0.20 to 0.28 and permeability values from 1 mD to 20 mD. The spiculite lithofacies is mostly misclassified as argillaceous fine-grained sandstone, showing similar porosity values. However, some of the measurements belonging to the argillaceous fine-grained lithofacies class exhibit higher porosity and permeability values, similar to the other fine-grained sandstones, and fall in the blue ellipsoid in Fig. 11.

Here, we see that even though most of the core measurements can be separated by the identified clusters in

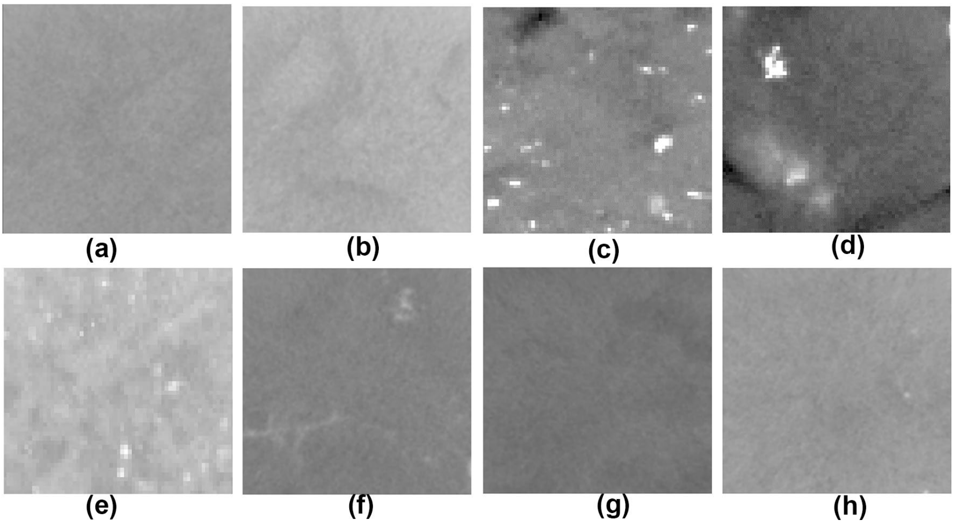


Fig. 10 Examples of very fine-grained lithofacies classes with similar textures and grain sizes with no distinct features. This type of images confuse the trained classifier and result in misclassifications and model deficiencies. **a** Marl, **b** Marl with caliche cementation, **c** Mudstone, **d** Mudstone with high-density minerals, **e** Cemented very fine-grained sandstone, **f** Massive very fine-grained green sandstone, **g** Muddy fine-grained sandstone, **h** Continental mudstone. The size of images is 64 × 64 pixels

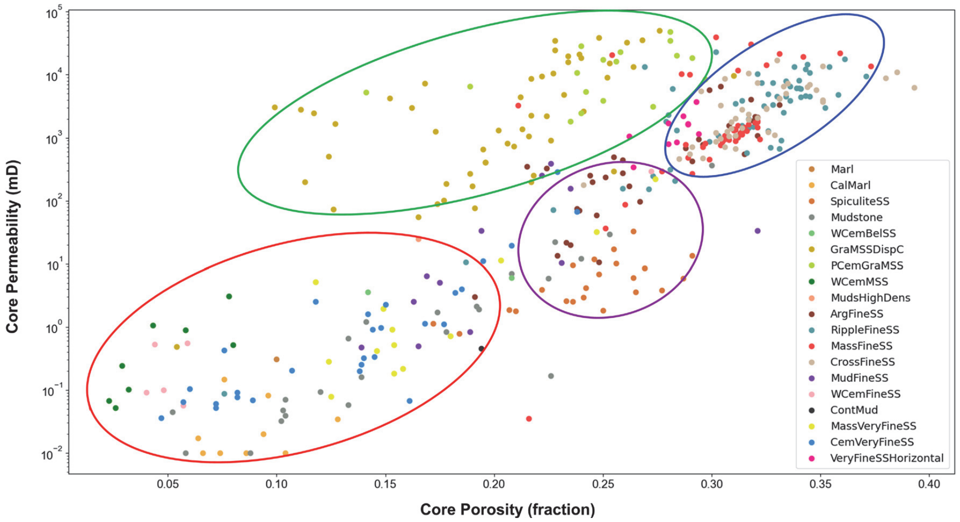


Fig. 11 Porosity–permeability cross-plot from available core measurements for the studied well. The lithofacies derived from core description are shown in different colors. The misclassified lithofacies exhibit similar porosity–permeability relationship marked by ellipsoids with different colors

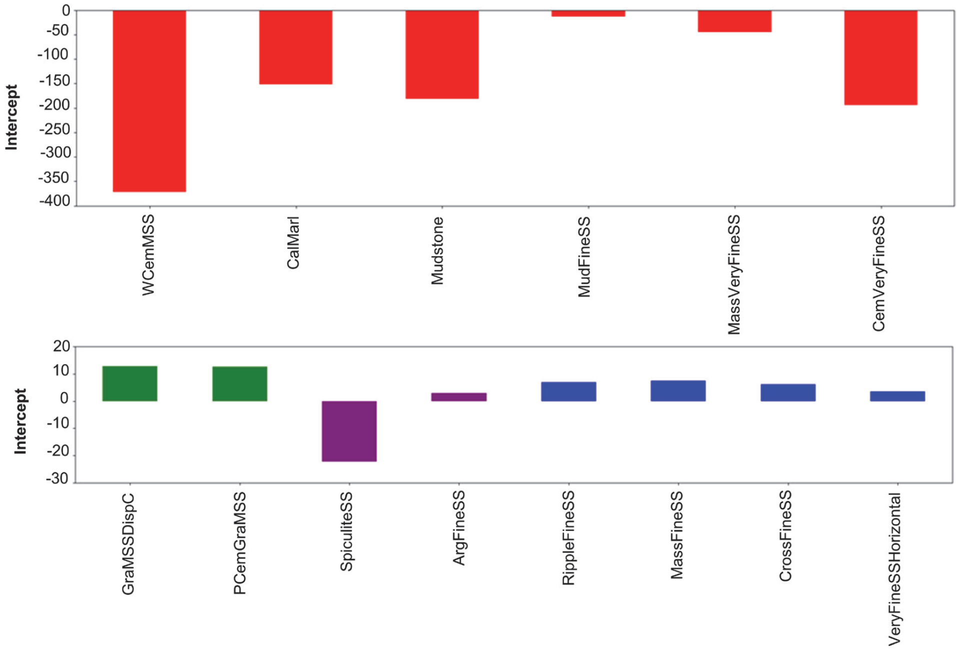


Fig. 12 Intercept values computed from the porosity–permeability cross-plot for misclassified lithofacies. Similar lithofacies, misclassified by the classifier, are presented with similar colors corresponding to the lithofacies clusters identified in Fig. 11

Fig. 11, some of the measurements exhibit a wide range of porosity and permeability values falling into more than one cluster, e.g., mudstone, cemented very fine-grained sandstone and argillaceous fine-grained sandstone samples.

In order to investigate the acquired classification results more quantitatively, we fit a log-linear regression line to map the porosity–permeability relationships of different lithofacies with more than five measurements. The computed intercepts can be used as an indication of similarity in transport properties between different lithofacies. The resulted intercept values are presented in Fig. 12; from this we can clearly infer that most of the lithofacies with similar transport properties tend to group into similar sets of intercept values. However, argillaceous fine-grained sandstone samples exhibit similar intercept values to the samples in the blue ellipsoid, which is expected due to the presence of argillaceous samples with higher range of porosity and permeability falling into the blue ellipsoid in Fig. 11. The colors in Fig. 12 correspond to the lithofacies clusters identified by ellipsoids in Fig. 11. Considering the similarities in transport properties of the misclassified lithofacies, it is not unreasonable to expect classification confusion amongst these classes.

In addition to the aforementioned similarities in texture and grayscale values, there are other issues that can create uncertainties and affect the training process and generalization capability of the trained classifier. One issue is related to the dipping and interchanging lithofacies. As an example, fine-grained argillaceous sandstones, ripple cross-laminated and cross-stratified sandstones interchange within the studied intervals creating difficulties in assigning a clear boundary during core description. Moreover, in the intervals with dipping lithofacies it is not easy to define a horizontal bed boundary. Another important point is related to the groundtruth labels derived from manual core description. These labels are assigned by visual inspection of the whole cores (or core photos), and they do not have pixel-wise resolution creating inconsistencies during training phase.

Figure 13 shows a section of the predicted test set together with the 2D whole core CT image and expert-derived core description. The classifier is able to predict the granule-rich (PCemGraMSS and GraMSSDispC) and well-cemented medium-grained sandstone lithofacies with fair accuracy. However, mudstone and fine-grained sandstone lithofacies (ripple cross-laminated and cross-stratified) are confused with other similar lithofacies.

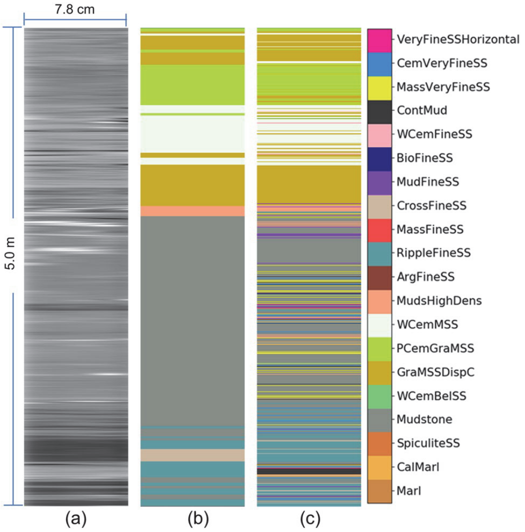


Fig. 13 Lithofacies prediction results from a section of the test set. **a** 2D whole core CT image, **b** Lithofacies classes from manual core description, **c** Lithofacies prediction using the trained CNN classifier

Table 4 Confusion matrix of four classes. Here, we consider four classes for simplicity

	GraMSS-DispC	PCem-GraMSS	RippleFineSS	CrossFineSS
GraMSS-DispC	0.92	0.02	0	0
PCem-GraMSS	0.26	0.71	0	0
RippleFineSS	0	0	0.53	0.19
CrossFineSS	0.02	0	0.42	0.44

6 Post-classification processing

In the previous section, we mentioned that the confusion matrix can provide invaluable information about the similarities and relationships between different lithofacies classes, and then we showed that the confused lithofacies classes exhibit similar porosity–permeability trends. In fact, in Fig. 11 we see that the misclassified lithofacies group into four different rock classes based on their porosity and permeability relationships. This guides us to consider if the lithofacies classification task can be coarsened with respect to the number of lithofacies classes. For this end inspired by Godbole [16], we use the information acquired from the confusion matrix to generate lithofacies hierarchies based on the degree of confusion for the different lithofacies classes.

6.1 Automatic generation of lithofacies hierarchies

Hierarchical clustering is a method in clustering analysis that aims at building a hierarchy of clusters based on a predefined similarity metric. Generally, two approaches are considered in performing hierarchical clustering analysis, i.e., *agglomerative* and *divisive clustering* [27]. Agglomerative clustering, also called the ”bottom-up” approach, starts with each element in a singleton cluster and pairs of clusters being merged successively until a specific stopping criterion is satisfied. The divisive, also called the ”top-down” approach, starts instead with all the elements in a single cluster; splitting is then performed recursively by moving down in the hierarchy. In hierarchical clustering similar clusters are grouped successively using a similarity metric, which is often a distance measure defined on the feature space [27]. The most common similarity metrics are Euclidean distance, Mahalanobis distance and Kullback-Leibler distance measure. There are different methods to measure distance between clusters; among these, the single-linkage [46], the complete-linkage [28], and the minimum variance (Ward) [51] methods are the most popular ones. More specifically, the single linkage (or nearest neighbor) clustering method looks for pairs of elements from two clusters that have minimum distance. In other words this approach basically considers recursively the closest pairs of elements from two clusters to measure the distance. In the complete linkage method, instead, the distance between two clusters is computed as the distance between the farthest elements of the two clusters. In both cases, the clusters with minimum distance measure are merged to form a larger cluster. The single-link algorithm is simple to implement, but it is known to suffer from chaining effects [35] that produce elongated clusters and long chains. By contrast, the complete link algorithm forces consistent diameter and spherical clusters. The Ward’s clustering method is then a special case of an objective function approach that looks for aggregate deviations of the elements. In fact, this method pretends to merge two clusters, and then estimates a centroid for the resulting cluster and calculates the sum of the squared deviations of all the elements from the new centroid. This algorithm then picks the merge with minimum within cluster variance or the merge with smallest deviation from the new centroid. The output of the hierarchical clustering is presented in a dendrogram representing the nested clustering of the elements and their similarity levels.

In this study, we perform hierarchical clustering using the empirical confusion matrix from the classifier as the

Table 5 Similarity matrix computed using the confusion matrix in Table 4

	GraMSS-DispC	PCem-GraMSS	RippleFineSS	CrossFineSS
GraMSS-DispC	0	1.35	1.66	1.78
PCem-GraMSS	–	0	1.69	1.81
RippleFineSS	–	–	0	0.38
CrossFineSS	–	–	–	0

quantitative measure of distance between the various lithofacies. This corresponds to use an Euclidean distance as the inter-class similarity metric between lithofacies class vectors in the confusion space. More precisely, the Euclidean distance is, in our work, calculated by summing up the absolute differences in the coordinate values of two class vectors. To exemplify the process, consider the confusion matrix in Table 4, where for simplicity we show only the results relative to four classes. Each class is represented by a vector in the confusion space, i.e., $\overrightarrow{GraMSSDispC} = \{0.92, 0.02, 0, 0\}$ represents GraMSSDispC lithofacies class. The Euclidean distances mentioned above are then calculated by summing up the absolute differences in the coordinate values of the class pairs. In this way it is possible to compute an upper triangular similarity matrix as the one shown in Table 5. This, in particular, clearly shows that RippleFineSS and CrossFineSS classes are the most similar ones among the set of classes considered in this sub-confusion matrix used to exemplify the process.

The computations and considerations in the example above are then performed and observed in the original complete confusion matrix; the resulting similarity matrix is then used as the input for the hierarchical agglomerative clustering step.

The dendrogram resulting from this clustering step is presented in Fig. 14 and shows the overall result of clustering similar lithofacies classes together. It is worth mentioning that various clustering methods result in different dendrogram structures. In this work, we started by performing hierarchical agglomerative clustering using all the three methods mentioned above, i.e., single-linkage, complete-linkage and Ward's method; we then observed that, among these approaches, the Ward's method returned the clustering structure that is the most coherent in terms of keeping lithofacies with similar transport properties together.

We also note that the vertical axis in a dendrogram is used as a reference distance that shows the similarity of the lithofacies classes. This means that the plot shows not only how different the classes are, but also the order by which lithofacies clustering occurs. We note that the obtained dendrogram clearly reflects the semantic similarity of the lithofacies classes in the confusion space. Indeed, for example, the plot shows that mudstone and massive very fine-grained green sandstone (MassVeryFineSS) classes are grouped before any other lithofacies classes; this is in line with the fact that these facies are, from a lithological perspective, the most similar ones within the set of classes we considered. The second most similar lithofacies classes are ripple cross-laminated (RippleFineSS) and cross-stratified fine-grained sandstone (CrossFineSS). As it should be, they form in the obtained dendrogram the second cluster in the hierarchy. The third cluster instead forms by merging the muddy fine-grained sandstones and well-cemented sandstones with Belemnite fossils. Then, this newly formed cluster is merged with the first cluster at a higher level of similarity distance. Moreover, argillaceous fine-grained sandstone (ArgFineSS) class clusters with spiculite sandstone. These lithofacies classes show a high degree of confusion with each other in the confusion space, as confirmed by Fig. 9.

As we explore the dendrogram upward, the similarity of lithofacies classes that are clustering together decreases. We indeed can note that the hierarchical clustering derived from similarity of lithofacies classes in the

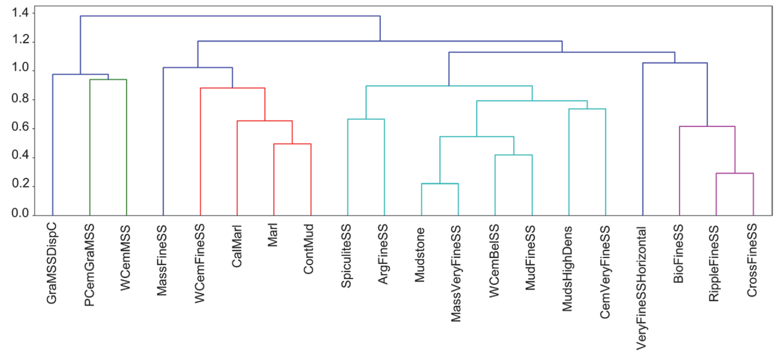


Fig. 14 Dendrogram of the process of clustering the lithofacies classes together using as a distance metric the confusion matrix that has been calculated by the proposed CNN classification algorithm

confusion space mostly results in grouping of lithofacies with similar grain sizes, textures and transport properties. However, as an example, we notice that the well-cemented medium-grained sandstone (WCemMSS) class is first merged with the poorly cemented granule-rich sandstone class (PCemGraMSS), and at a slightly higher level they merge with granule-rich sandstone with dispersed cementation (GraMSSDispC). Recall then that it was previously shown that the granule-rich lithofacies core measurements spread out in the regions with high permeability values ranging from 30 mD up to 50 Darcy (i.e., the green ellipsoid in Fig. 11), where the majority of samples exhibit porosity and permeability values above 0.20 and 1 Darcy, respectively. On the other hand, the well-cemented sandstone samples (WCemMSS) are characterized by porosity and permeability values less than 0.10 and 5 mD, respectively. Therefore, merging these classes, with completely different transport properties, does not seem reasonable.

6.2 Lithofacies prediction using the Coarsened CNN classifier

As mentioned in the previous subsection, the current lithofacies classification task can be coarsened with respect to the number of classes by merging similar misclassified lithofacies classes. More specifically, based on the porosity–permeability relationships and hierarchical clustering results, we propose grouping the lithofacies classes into four rock classes, as presented in Table 6.

Following this classification, the groundtruth labels derived from manual core description can be modified so to reflect the four superclasses above instead of the original 20 ones. This implies that one can retrain the original CNN classifier proposed above using this new set of labels and also perform a new round of testing. The resulting confusion matrix is shown in Fig. 15, from which we see that the classifier is able to predict rock classes 1, 2, and 4 with high recall values.

However, rock class 3 is still predicted with a relatively low recall (0.65), and it is mostly confused with rock classes 1 and 4. To inspect why, consider the corresponding rock classes, shown in Fig. 16. The plot shows that the coarsened classifier generalizes well and predicts individual rock classes with high accuracy. The classifier even shows higher pixel-wise precision in detecting thin layers and bed boundaries to the point that it is able to detect thin layers that are not picked by the manual core description. As an example, in Fig. 16, the 2D CT image cross section shows a clear change in the gray scale values in the section marked by the green rectangle in Fig. 16A. Here, we see that the more porous and permeable layer (characterized by darker grayscale values) is underlain by a tighter layer marked by the red rectangle. The tight layer is characterized by brighter gray scale values compared to the layers above and below, but this was not picked during manual core description. At the same time, this layer is accurately detected by the CNN classifier. More investigation of this

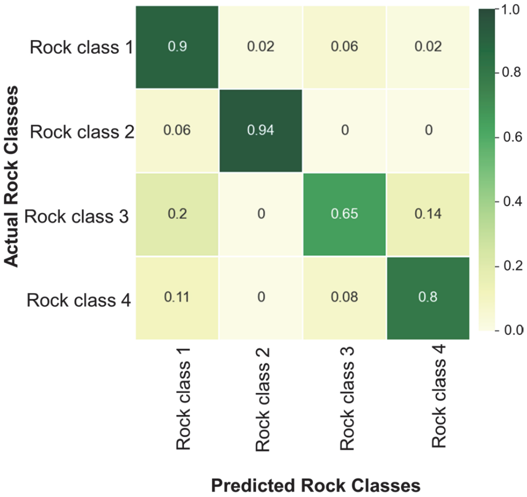


Fig. 15 Confusion matrix on the test set using the coarsened classifier, where the original 20 lithofacies classes are merged into four rock classes

Table 6 Proposed rock classes resulted from merging similar lithofacies classes

Rock classes	Clustered lithofacies	Description
Rock class 1	Marl, CalMarl, ContMud, WCemFineSS, Mudstone, MudsHighDens, MudFineSS, WCemBelSS, WCemMSS, CemVeryFineSS, MassVeryFineSS	Very fine- to medium-grained sandstones, well-cemented very fine- to medium-grained sandstones, marl and mudstones
Rock class 2	GraMSSDispC, PCemGraMSS	Medium-grained granule-rich sandstones, poorly cemented / with dispersed calcite cementation
Rock class 3	SpiculiteSS, ArgFineSS	Fine-grained spiculite sandstones and fine-grained argillaceous sandstones
Rock class 4	RippleFineSS, CrossFineSS, BioFineSS, MassFineSS, VeryFineSSHorizontal	Fine-grained sandstones with different types of laminations

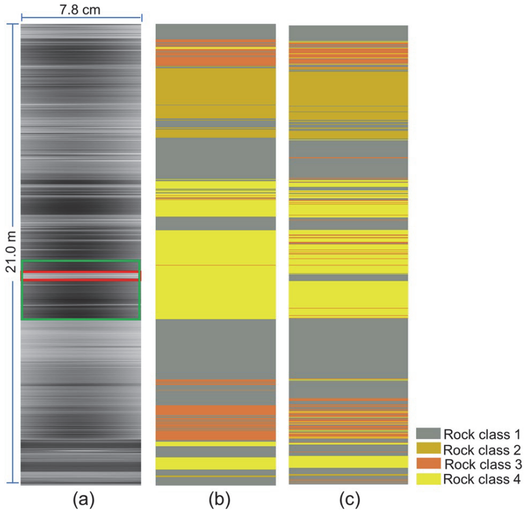


Fig. 16 Predicted rock classes on the test section of the well (approximately 21 meters) (**c**), shown with actual rock classes (**b**) and the 2D cross section of the input CT images (**a**). The scaled-up classifier is predicting the rock classes with high accuracy

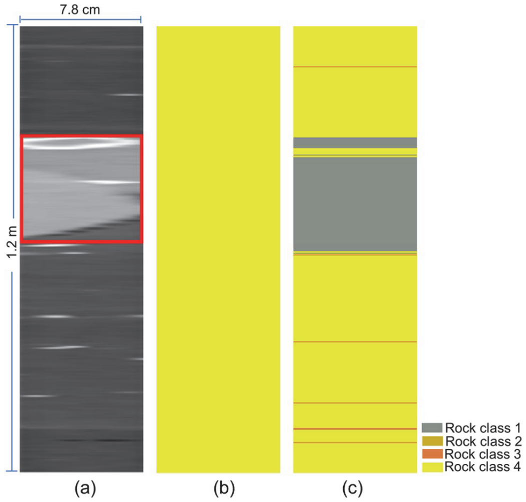


Fig. 17 Zoomed interval of the test set (approximately 1.2 meters), where the CNN classifier is able to pick the calcite nodule. **a** 2D cross section of the input CT images, **b** Rock classes from manual core description, **c** Predicted rock classes using the scaled-up classifier

interval reveals that the tight layer is actually a big calcite nodule encapsulated within the massive fine-grained sandstone lithofacies (Fig. 17). This calcite nodule is classified as rock class 1 that contains lithofacies classes with high amount of calcite cementation, most probably due to similar grayscale values.

7 Conclusions

In this study, the capability of CNN to classify lithology, based on the 2D whole core CT image slices, was investigated, and its performance was characterized in detail.

A CNN classifier was trained to learn features associated with 20 various lithofacies classes derived from manual core descriptions. The trained classifier was then used to predict lithofacies on the unseen test set images.

The preliminary results revealed that the trained classifier showed lithofacies-dependent performance and it misclassified, to various degrees, specific lithofacies classes with similar grain size, gray-scale values, and transport properties.

The obtained prediction confusion matrix was then utilized as a valuable tool to understand the performance limits of the CNN classifier and to combine the similar lithofacies into rock classes using an automatic hierarchical clustering approach.

Applying the CNN classifier on these clustered classes shows that the new approach generalizes well and predicts the rock classes with high recall values. Moreover, it shows higher pixel-wise precision, in detecting thin layers, compared to expert-derived core description, thereby providing higher resolution information than the one extracted during the manual labeling process.

The proposed classifier is trained based on data from a single well with imbalanced distribution of lithofacies classes. This might result in lower prediction performance on the classes with lower proportions. Adding more training images for those classes, preferable from other wells with similar lithology, might have a positive impact on the performance of the classifier.

As expected, uncertainties associated with manual core description, interchanging and dipping lithofacies can also affect the training process and generalization capability of the trained classifier.

It is worth to mention that the network architecture might affect the results, but it is not expected to change the conclusions in this study. For comparison purposes, the VGG16 architecture [45] was tested out and its performance was compared with the proposed architecture. However, this change of the CNN architecture had minor impact on the acquired results.

Acknowledgements This research is a part of BRU21 – NTNU Research and Innovation Program on Digital and Automation Solutions for the Oil and Gas Industry (www.ntnu.edu/bru21) supported by Equinor in Norway. Carl Fredrik Berg acknowledges support from the Research Council of Norway through its Centre of Excellence funding scheme with Project No. 262644. We would like to thank Equinor for providing the data and the permission for publishing this work.

Funding This study was supported by Equinor.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Availability of data and material Due to confidentiality agreements, the data supporting the findings of this study cannot be made available.

Disclaimer The views and opinions expressed in this paper are those of the authors and do not necessarily reflect those of the Joint Industry Research Program members or example data owners.

References

1. Al-Anazi A, Gates I (2010a) On the capability of support vector machines to classify lithology from well logs. Nat Resour Res 19(2):125–139
2. Al-Anazi A, Gates I (2010b) A support vector machine algorithm to classify lithofacies and model permeability in heterogeneous reservoirs. Eng Geol 114(3–4):267–277
3. Al-Obaidi M, Heidari Z, Casey B, Williams R, Spath J et al (2018) Automatic well-log-based fabric-oriented rock classification for optimizing landing spots and completion intervals in the midland basin. Society of Petrophysicists and Well-Log Analysts
4. Anjos CE, Avila MR, Vasconcelos AG, Neta AMP, Medeiros LC, Evsukoff AG, Surmas R, Landau L (2021) Deep learning for lithological classification of carbonate rock micro-ct images. Comput Geosci 25(3):1–13
5. Ball K, Arbus T, Odi U, Sneed J (2017) The rise of the machines, analytics, and the digital oilfield: Artificial intelligence in the age of machine learning and cognitive analytics. Unconventional Resources Technology Conference
6. Baraboshkin EE, Ismailova LS, Orlov DM, Zhukovskaya EA, Kalmykov GA, Khotylev OV, Baraboshkin EY, Koroteev DA (2020) Deep convolutions for in-depth automated rock typing. Comput Geosci 135:104330
7. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(1):281–305
8. Bize-Forest N, Lima L, Baines V, Boyd A, Abbots F, Barnett A et al (2018) Using machine-learning for depositional facies prediction in a complex carbonate reservoir. Society of Petrophysicists and Well-Log Analysts, pp 1–11
9. Ceci M, Hollmén J, Todorovski L, Vens C, Džeroski S (2017) Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, vol 10535. Springer, Berlin
10. Chawshin K, Gonzales A, Berg CF, Varagnolo D, Heidari Z, Lopez O (2021) Classifying lithofacies from textural features in whole-core ct-scan images. SPE Reserv Eval Eng 24(02):341–357
11. Chollet F (2015) Keras: The python deep learning library. Astrophysics Source Code Library: ascl-1806
12. De Lima R, Suriamin F, Marfurt KJ, Pranter MJ (2019a) Convolutional neural networks as aid in core lithofacies classification. Interpretation 7(3):SF27–SF40
13. De Lima RP, Bonar A, Coronado DD, Marfurt K, Nicholson C (2019b) Deep convolutional neural networks as a geological image classification tool. Sediment Rec 17:4–9
14. De Lima RP, Duarte D, Nicholson C, Slatt R, Marfurt KJ (2020) Petrographic microfacies classification with deep convolutional neural networks. Comput Geosci 142:104481
15. Dubois MK, Bohling GC, Chakrabarti S (2007) Comparison of four approaches to a rock facies classification problem. Comput Geosci 33(5):599–617
16. Godbole S (2002) Exploiting confusion matrices for automatic generation of topic hierarchies and scaling up multi-way classifiers. Annual Progress Report. Indian Institute of Technology, Bombay, India
17. Gonzalez A, Kanyan L, Heidari Z, Lopez O et al (2019) Integrated multi-physics workflow for automatic rock classification and formation evaluation using multi-scale image analysis and conventional well logs. Society of Petrophysicists and Well-Log Analysts
18. Hall B (2016) Facies classification using machine learning. Lead Edge 35(10):906–909
19. Hall BJ, Govert A, Energy C (2016) Techniques for Using Core CT Data for Facies Identification and Analysis. SPE/AAPG/SEG Unconventional Resources Technology Conference
20. Hinton G, Srivastava N, Swersky K (2012) Neural networks for machine learning lecture 6a overview of mini-batch gradient descent
21. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580
22. Horrocks T, Holden EJ, Wedge D (2015) Evaluation of automated lithology classification architectures using highly-sampled wireline logs for coal exploration. Comput Geosci 83:209–218
23. Hutter F, Hoos HH, Leyton-Brown K (2011) Sequential model-based optimization for general algorithm configuration. In: International Conference on Learning and Intelligent Optimization. LION 2011: Learning and Intelligent Optimization, pp 507–523
24. Ian Goodfellow YBaAC (2016) Deep Learning. MIT Press.
25. Imamverdiyev Y, Sukhostat L (2019) Lithological facies classification using deep convolutional neural network. J Petrol Sci Eng 174:216–228

26. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of ICML, pp 448–456
27. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM computing surveys (CSUR) 31(3):264–323
28. King B (1967) Step-wise clustering procedures. J Am Stat Assoc 62(317):86–101
29. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. In: International Conference on Learning Representations
30. Kirichev M, Slavov T, Momcheva G (2021) Fuzzy u-net neural network architecture optimization for image segmentation. IOP Conf Ser Mater Sci Eng 1031(1):012077
31. Korjani M, Popa A, Grijalva E, Cassidy S, Ershaghi I et al (2016) A new approach to reservoir characterization using deep learning neural networks. Presented at the SPE Western Regional Meeting, Anchorage, Alaska, USA, SPE-180359-MS
32. Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2017) Hyperband: A novel bandit-based approach to hyperparameter optimization. J Mach Learn Res 18(1):6765–6816
33. Malki H, Baldwin J, Kwari M et al (1996) Estimating permeability by use of neural networks in thinly bedded shaly gas sands. SPE Comput Appl 8(02):58–62
34. Mustra M, Delac K, Grgic M (2008) Overview of the dicom standard. In: 50th International Symposium ELMAR, 1, pp 39–44
35. Nagy G (1968) State of the art in pattern recognition. Proc IEEE 56(5):836–863
36. Odi U, Nguyen T (2018) Geological facies prediction using computed tomography in a machine learning and deep learning environment. Unconventional Resources Technology Conference, Society of Exploration Geophysicists, pp 336–346. URTEC-2901881-MS
37. O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L (2019) Keras Tuner. https://github.com/keras-team/keras-tuner
38. Rafik B, Kamel B (2017) Prediction of permeability and porosity from well log data using the nonparametric regression with multivariate analysis and neural network, hassi r'mel field, algeria. Egypt J Petrol 26(3):763–778
39. Renter JA (1989) Applications of computerized tomography in sedimentology. Marine Georesour Geotechnol 8(3):201–211
40. Rogachev A, Melikhova E (2020) Automation of the process of selecting hyperparameters for artificial neural networks for processing retrospective text information. IOP Conf Ser Earth Environ Sc 577:012012
41. Rogers S, Chen H, Dt K-M, Fang J (1995) Predicting permeability from porosity using artificial neural networks. AAPG Bulletin 79(12):1786–1797
42. Rogers SJ, Fang J, Karr C, Stanley D (1992) Determination of lithology from well logs using a neural network. AAPG Bulletin 76(5):731–739
43. Schneider CA, Rasband WS, Eliceiri KW (2012) Nih image to imagej: 25 years of image analysis. Nat Methods 9(7):671–675
44. Scholkopf B, Smola AJ (2018) Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning series, MIT Press, Cambridge MA
45. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR)
46. Sneath PH, Sokal RR et al (1973) Numerical taxonomy. The principles and practice of numerical classification. W. H. Freeman, San Francisco, CA
47. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, pp 2960–2968
48. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
49. Tschannen V, Delescluse M, Rodriguez M, Keuper J (2017) Facies classification from well logs using an inception convolutional network. Computer Vision and Pattern Recognition
50. Valentin MB, Bom CR, Coelho JM, Correia MD, Márcio P, Marcelo P, Faria EL (2019) A deep residual convolutional neural network for automatic lithological facies identification in brazilian pre-salt oilfield wellbore image logs. J Petrol Sci Eng 179:474–503
51. Ward JH Jr (1963) Hierarchical grouping to optimize an objective function. J Am stat Assoc 58(301):236–244
52. Wong P, Henderson D, Brooks L et al (1998) Permeability determination using neural networks in the ravva field, offshore india. SPE Reserv Eval Eng 1(02):99–104
53. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9(4):611–629

Paper III

This paper is not included due to SPE copyright retrictions

Paper IV

This paper is awaiting publication and is not included in NTNU Open