CBPF – Centro Brasileiro de Pesquisas Físicas

Dissertação de Mestrado

Deep Learning Methods on Geological Reservoir Borehole Log Images and Applications

Manuel Blanco Valentín

Orientador
Dr. Márcio Portes de Albuquerque
Co-Orientador
Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2018

Manuel Blanco Valentín

Deep Learning Methods on Geological Reservoir Borehole Log Images and Applications

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física com Ênfase em Instrumentação Científica.

CBPF – Centro Brasileiro de Pesquisas Físicas

Supervisor: Dr. Márcio Portes de Albuquerque
Co-supervisor: Dr. Clécio Roque De Bom

Rio de Janeiro, RJ
2018

Blanco Valentín, Manuel
Deep Learning Methods on Geological Reservoir Borehole Log Images and Applications/ Manuel Blanco Valentín. - 2018
179 f. : il.

Dissertação de Mestrado – CBPF – Centro Brasileiro de Pesquisas Físicas , Rio de Janeiro, RJ, 2018.
Supervisor: Dr. Márcio Portes de Albuquerque

1. xxxxx. 2. xxxxx. 3. xxxxx. 4. xxxxx. I. Título
CDU 02:141:005.7

Manuel Blanco Valentín

Deep Learning Methods on Geological Reservoir Borehole Log Images and Applications

Trabalho apresentado ao Programa de Pós-Graduação no Centro Brasileiro de Pesquisas Físicas como requisito parcial para obtenção do grau de Mestre em Física com Ênfase em Instrumentação Científica.

Approved work. Rio de Janeiro, RJ, __th March 2018:

Dr. Márcio Portes de Albuquerque
Supervisor

Clécio R. De Bom
Co–supervisor

Professor

Professor

Professor

Rio de Janeiro, RJ
2018

Dedico este trabalho à minha esposa, Thamires Bengaly, e à minha família.

Acknowledgements

First of all I would like to thank both my supervisor *Márcio P. de Albuquerque* and co–supervisor *Clécio R. De Bom* for making this work possible, as well as for all kind suggestions, corrections and hints, which greatly increased the quality of this work. Without their effort and their professionality and knowledge this work would never have been possible.

I would also like to thank PETROBRAS and the Brazilian Government for promoting the type of scientific projects that resulted on this work, which has been not only a great opportunity for my career, but also improves the scientific quality level of Brazil and the whole academia with the promoted research.

I would like to thank both PETROBRAS and ANP for providing to us the petrophysical data without which this work would definetely have not been possible.

I would also like to heartfully thank my wife for all her support and understanding, for being the perfect partner and for even having to stand some philosophical arguments about this thesis and the project I took part on for the last three years.

I thank my family and the family of my wife for all their support and understanding, and for being there when we most needed them. I want to take this opportunity to thank *Maury D. Correia*, *Anelise Lima*, *Candida Menezes*, *André Compan*, *Ana Patrícia Cavalcante*, *Lenita Fioriti* and everyone at PETROBRAS that helped me throughout this project, as well as all my project partners: *Luciana*, *Felipe*, *Juliana* and *André*;*Marcelo P. de Albuquerque* and the whole CBPF staff.

"Not until we are lost do we begin to understand ourselves" – Henry David Thoreau

Abstract

Reservoir Characterization is a major field of petrophysics and geology which aims to obtain empirical and theoretical models for an accurate description of the factors and forces that create and modify a Hydrocarbon Reservoir, as well as the different physical properties that might characterize and define it properly by mapping different physical properties of the reservoir soil. Amongst all properties, three are considered key when determining whether it is feasible or not to exploit a reservoir: saturation of phases, permeability and porosity of the rock.

Obtaining accurate permeability and effective porosity measurements of the reservoir soil is not an easy task as it usually involves obtaining undamaged core samples from the reservoir itself, to which different expensive and destructive laboratory tests are applied. From this perspective, several authors have proposed to use borehole wirelogs and seismic data to estimate different geological properties such as permeability and porosity logs, rock facies or electrofacies.

In this work, we present two novel methods that can be applied to borehole image logs in order to extract accurate estimations of petrophysical and geological properties from them. The first one, which is based on deep stacked autoencoders, uses ultrasonic and microresistivity borehole image data to obtain an estimation of both permeability and porosity logs. We tested our method on borehole data obtained from a carbonate reservoir in the brazilian pre–salt region, achieving an accuracy of over 96%. The second novel method we propose in this work is based on deep convolutional neural networks, which extract high–order inner features of ultrasonic and microresistivity borehole image logs in order to predict rock reservoir facies. We also tested this method in another carbonate reservoir in the brazilian pre–salt region, achieving a global accuracy of over 97%.

These two methods address the problem of petrophysical properties estimation from the Deep Learning perspective. The fact that we use sliding windows and/or convolutional techniques allows our methods to extract properties from the Borehole Images by retaining spatial surrounding data, rather than just looking at single data points (as most of the previous benchmark did). Our results, along with the results obtained in previous work in the field, suggest that there exists indeed a link between the properties captures on these images and petrophysical logs.

Apart from that, we also present three novel methods to enhance/filter and process borehole image logs. The first of them is an automatic method for detecting breakouts. Characterizing these structures in borehole data is important as it helps understanding the stresses that took place while drilling. Currently, breakouts are usually selected manually by approximative methods, thus leading to inaccurate results. Our method, which is based on Transit–Time Borehole Image Data, has shown high accuracy in characterizing the geometry of these formations automatically for the tested data. The second novel

method for processing borehole images aims to eliminate the distorsion caused by the distance between an uncentered probe and the borehole wall when obtaining some borehole image logs, such as the ultrasonic one. In our method, we use transit–time image logs to estimate the distance between the probe and the borehole wall and remove the distorsion caused by this effect. Lastly, we present a method for automatic pad artifact removal from borehole image logs. Our method is based on a specific pad–reject function to the ultrasonic data, so that the resulting image contains only the lithological information of the wellbore that was had been masked by the artifacts, which might help geologists in any further analysis of this data.
Key-words: Deep Learning, Convolutional Neural Network, Borehole Image Logs, Reservoir, Oil, Image Processing.

Resumo

A Caracterização de Reservatórios é um campo importante da petrofísica e da geologia que tem por objetivo obter modelos teóricos e empíricos capazes de descrever de forma precisa os fatores e forças que criam e modificam os reservatórios de hidrocarbonetos, assim como as diversas propriedades físicas que possam caracterizá-lo e definí-lo, mediante o mapeamento de diversas propriedades físicas do reservatório. Dentre todas as propriedades, três são consideradas chave para determinar a viabilidade da exploração de um reservatório: a saturação de fases, a permeabilidade e a porosidade da rocha.

Obter medidas confiáveis de permeabilidade e porosidade efetiva do reservatório não é uma tarefa simples, já que geralmente implica a necessidade de obter amostras rochosas intatas do próprio reservatório, nas quais diversos testes destrutivos e caros são aplicados. Desde essa perspectiva, diversos autores propuseram usar curvas de poços e dados sísmicos para estimar diferentes propriedades geológicas tais como permeabilidade, porosidade, fácies de rochas ou eletrofácies.

Neste trabalho, apresentamos dois novos métodos que podem ser aplicados a perfis de imagens de poços para extrair deles estimativas precisas de propriedades geológicas e petrofísicas. O primeiro, que é baseado em *Deep Stacked Autoencoders*, usa imagens de poço ultrassônicas e de microresistividade para obter uma estimativa de curvas de permeabilidade e porosidade efetiva. Testamos nosso método em dados originários de um reservatório de carboneto do pré–sal, obtendo uma precisão na estimativa de mais de 96%. O segundo método que apresentamos neste trabalho é baseado em redes neurais convolucionais profundas, as quais são capazes de extrair características internas sofisticadas de imagens de poço ultrassônicas e de microresistividade, para poder estimar fácies de rochas. Esse método também foi testado com dados de um outro poço do pré–sal, obtendo uma precisão global de estimativa de mais de 97%.

Estes dois métodos abordam o problema da estimativa de propriedades petrofísicas desde o ponto de vista do deep learning. O fato de usarmos janelas deslizantes e/ou técnicas convolutivas permite que nossos métodos extriam propriedades das Imagens de Poços que mantém as características espaciais das redondezas de cada amostra, ao invés de tratar cada amostra como um ponto único (como a maioria dos métodos anteriores analisados faziam). Nossos resultados, junto com os resultados obtidos em trabalhos prévios ao nosso nessa área, sugerem a existência de uma ligação entre as propriedades capturadas por essas imagens e estas curvas petrofísicas.

Por outro lado, nós também apresentamos três novas técnicas para realçar/filtrar e processar imagens de poços de reservatórios. A primeira delas é um método de detecção automática de breakouts. Caracterizar estas estruturas em dados de reservatório é importante porque ajuda a entender a distribuição de estresses no poço durante a perfuração. Atualmente, os breakouts são detectados manualmente por métodos aproximados, o que

acaba gerando resultados pouco precisos. Nosso método, que é baseado em imagens de reservatório de tempo de transito, mostrou uma alta precisão na caracterização automática destas formações nos dados testados. O segundo método de processamento de imagens de poço tem por objetivo eliminar a distorção causada pela distância entre sensores não centralizados e a parede do poço, ao obter imagens de poço tais como as ultrassônicas. Nosso método usa imagens de tempo de trânsito para estimar essa distância e remover a distorção causada por este efeito. Por último, nós apresentamos um método para remoção de artefatos de patins de imagens de poço de reservatório. Nosso método se baseia na aplicação de uma função específica de rejeição de patins em dados ultrassônicos, de forma que a imagem resultante contém unicamente a informação litológica do poço que foi mascarada pelos artefatos, o que pode ajudar aos geólogos em futuras possíveis análises do dado.

Palavras Chaves: Deep Learning, Rede Neural Convolucional, Perfil de Poço, Reservatório, Óleo, Processamento de Imagens.

List of Figures

List of Tables

List of abbreviations and acronyms

AAPG	American Association of Petroleum Geologists
ABI	Acoustic Borehole Imager
AE	Autoencoder
AI	Artificial Intelligence
ANFIS	Adaptive Neuro Fuzzy Interference System
ANN	Artificial Neural Network
ANP	Agência Nacional de Petróleo
BNN	Backpropagation Neural Network
BNORM	Batch Normalization Layer of a Convolutional Neural Network
BTA	Borehole Texture Analyzer
CNN/ConvNet	Convolutional Neural Network
CT	Computerized Tomography
CDT	Conductivity Wireline Log
ConvLayer	Convolutional Layer of a Convolutional Neural Network
DAE	Deep Autoencoder
DBN	Deep Belief Network
DDBN	Discriminative Deep Belief Network
DDN	Discriminative Deep Network

DIF	Drilled Induced Fractures
DNN	Deep Neural Network
DT	Sonic Wireline Log
EMI	Electrical Micro-Imaging Tool
FC	Fully–Connected Layer of a Convolutional Neural Network
FFNN	Feed-forward Neural Network
FMI	Fullbore Formation Micro Imager
FMS	Formation Micro-Scanner Log Data
GAN	Generative Adversarial Network
GR	Gamma Ray Response Wireline Log
HMM	Hidden Markov Model
HRAT	High Resolution Acoustic Televiewer
kNN	k–Nearest Neighbour Classifier
LLD	Deep Laterolog Wireline Logging Tool
LLS	Shallow Laterolog Wireline Logging Tool
MLP	Multilayer Perceptrons
NMR	Nuclear Magnetic Ressonance
NPHI	Neutron Porosity Response Wireline Log
OBMI	Oil-Based Microimager
PERM	Permeability
PHIE	Effective Porosity
PHIT	Total Porosity
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
RHOB	Bulk Density Response Wireline Log
ROC	Receiver Operating Curve

SLFNN	Single Hidden Layer Feedforward Neural Network
SP	Spontaneous Potential Response Wireline Log
SPHI	Sonic Porosity Wireline Log
SVM	Support Vector Machine
UBI	Ultrasonic Borehole Imager
VAE	Variational Autoencoder
WS	Water Saturation Log
ZSS	Zonal Subdivision Specification

Contents

Chapter 1

Introduction

The term Petrophysics[1] (from the Greek πέτρα, rock and φίσις, nature or physics –modern science–) is used to refer to the field of science that applies geology, physics, chemistry and mathematical models to study and characterize the properties of reservoir rocks (Buryakovsky et al., 2012). During this process, which is usually called reservoir characterization, petrophysics aims to achieve a detailed description regarding the composition and disposition of each layer of the reservoir, along with the structures found inside them and their physical properties, as well as to obtain empirical and theoretical models that might explain the geological processes that built itself.

The information regarding the physical properties of the reservoir obtained by the reservoir characterization process does not only help geologists understand how a reservoir was formed, but it also helps engineers, physicists and analysts to create models to predict how the rocks that form the reservoir might react to certain fluids flowing through them (Lucia, 2007). These models are critical to evaluate whether it is feasible or not to even consider that reservoir for hydrocarbon exploitation. The formations known as the brazilian pre-salt[2] is a clear example of this: even though it has been estimated that the total volume of hydrocarbons hidden under them can range from 50 to over 300 billion barrels of exploitable oil (Jones et al., 2011), engineers and technicians are still striving up to this day to find an efficient and reliable way to overcome the difficulties and challenges that the salt layer presents for any large-scale production proposed (da Costa Fraga et al., 2015).

Even when dealing with more accessible reservoirs, characterizing the field is a challenging task that requires both a high technical and economical effort. The most common data sources are:

[1] The term *Petrophysics* was coined by G.E.Archie and J.H.Thomeer in the early XX century (Thomas, 1992) and has been used since then to refer to the study of physical properties of rocks in oil/gas reservoirs.

[2] From portuguese *pré-sal*, which refers to a thick layer of salt that separates the reservoir rocks from the upper layers of sand at the bottom of the sea in the brazilian east–coast basins.

1. **Seismic data**: this source of data essentially shows the different densities and velocities of each one of the layers that compose the rocks of the reservoir. This data is often used to get an idea of where the hydrocarbons are most likely located (if any). Further detail about this data will be explained in Section 2.2.

2. **Wirelog data**: this source of data is obtained by drilling the reservoir in order to extract the most amount information from them, while causing the minimal impact on the field. Once these spots are drilled (and commonly, meanwhile they are drilled, as shown by Zhang et al. (2006)), several tools are introduced in the boreholes to gather information about the rocks that compose the reservoir. The information obtained during this process can be either represented by simple curves (i.e., gamma ray, neutron porosity or bulk density logs, to name a few), spectra logs (i.e., NMR decay times per depth) or images (i.e., acoustic and electrical borehole formation image data, like the ones on which this work is mainly focused). This data will be explained in more detail in Section 2.3 and Section 2.4.

3. **Core Analysis**: undamaged cores of the drilled boreholes are extracted[3] and taken to the laboratory where chemical tests are run on them. These tests have the purpose of obtaining specific petrophysical properties from the reservoir beddings, such as saturation, permeability, porosity or density that can only be obtained by destructive or direct laboratory experiments. The information extracted from these tests is essential in the process of reservoir characterization and production feasibility evaluation, as will be explained in detail in Section 2.5.

4. **Formation Tests**: this source of data help analysts to have a notion of how the reservoir can react while exploitation. Several small tests are run in order to extract a certain amount of hydrocarbon from the reservoir. Commonly small amounts of hydrocarbon are extracted (and usually discarded), as the main objective of formation tests is to gather direct information about the exploitability and the behaviour of the reservoir and their fluids while hydrocarbon extraction.

When all this information has been gathered, experts analyze it carefully, usually validating empirical models using several different production conditions (Dahaghi et al., 2010; Brouwer et al., 2004; Oldenburg et al., 2001; Burshears et al., 1986), and decide whether the field is exploitable or not. Generally, this process is developed during several months or even years. It is worth noticing that despite how important reservoir characterization is, there has not been found yet a unique automated procedure to obtain a single

[3]The proportion of rock core that is extracted to be used in the laboratory depends on different logistic factors. In order to extract intact cores it is required to use a special drill while drilling with high precaution. This process is very time and effort consuming and so it is common to retrieve only about 10 to 20% of the whole borehole core to be taken to the lab while discarding the rest of it (drilling it faster, with a much more destructive drill).

definitive objective prediction about the reservoir feasibility from all the field information retrieved. Instead, this process requires that a group of highly qualified and experienced experts analyze all that information in order to come up with a decision (Chopra and Marfurt, 2007; Doyen, 2007; Oliver et al., 2008; Kelkar et al., 2002).

In this context, the application of machine learning and artificial intelligence has proved to be exceptionally beneficial for the oil/gas industry, as will be detailed in Section 4, Section 5 and Section 6. As soon as computational methods increased their performance and started to be considered as more reliable methodologies to be applied on scientific and technical problems, they started to be used for reservoir characterization and simulation as well. Early works used computers to run complex reservoir flow simulations (Du et al., 1986), predict petrophysical parameters (Angeleri and Carpi, 1982) and used them for reservoir characterization purposes (Mohaghegh et al., 1994). Once these techniques became more and more sophisticated and specially once artificial neural networks began to be widely used in scientific applications, a more robust panorama for automatic analysis was developed, thus improving the performance in automatic reservoir characterization (Anifowose and Abdulraheem, 2011; Ouahed et al., 2003; Ouenes, 2000; Mohaghegh et al., 1996).

One major revolution in the digital processing field took place in the last decade with the invention of the convolutional neural networks (LeCun et al., 1998; Krizhevsky et al., 2012). These networks are a type of feed-forward[4] artificial neural network that use recursive filtering to extract inner local features from images that can later be used for regression, classification or characterization purposes on almost any kind of input data.

Deep convolutional networks are mainly used in data that has some sort of distribution: either spatial (i.e., images) or temporal (i.e., signals over time). They have proved to be a very effective methodology for extracting the internals patterns of data that help in the signal characterization process. Some examples of applications of convolutional networks (henceforth, ConvNets or CNNs), published recently, include using them in medicine to segment cancerous cells from brain images (Havaei et al., 2017) or from CT scans (Todoroki et al., 2017); in astrophysics to predict the presence of supernovae (Kimura et al., 2017) or gravitational lenses (Metcalf et al., 2018; Hezaveh et al., 2017) in retrieved images from space; in computer science to recognize faces automatically (Yan et al., 2017) or even the emotions contained in a certain conversation or speech (Mukaihara et al., 2017).

While borehole image logs have been commonly used for structure and object detection (i.e., breakouts, fractures, bedding planes), in the past decade some authors have used them for more sophisticated workflows than structures detection. Among them it is worth naming two, on which this thesis is focused: rock facies classification and flow

[4] A feed-forward neural network –henceforth, FFNN– is a type of neural network where the weights of each inner layer only affect the weights of the following layers, without any type of recursive loop.

petrophysical properties estimation. Different methodologies have been proposed using different types of data sources, for borehole lithology characterization using well images such as: applying wavelet filters to formation microscanner data (henceforth, FMS) and then extracting haralick textural features (Haralick et al., 1973) from the filtered images and using a discriminant analysis for facies classification, obtaining a success in the classification of 98% (Linek et al., 2007); using textural features extracted directly from FMS data and inputing them into a linear discriminant analysis (henceforth, LDA) and obtaining a classification result of 80% of success (Jungmann et al., 2011); or extracting textural features from FMS and dipmeter data and combining them with conventional wireline logs to use in an artificial neural network to both classify the facies in the well and obtain predictions of permeability logs (Russell et al., 2002). More recently, C. Menezes et al. de Jesus et al. (2016) have used acoustic image log data to estimate permeability and porosity logs from them using multi-level segmentation and non-linear optimization, achieving high quality results in their prediction.

In this work, we explore two different image processing fields. The first field, which is studied in Section 3, regards the application of modern image processing tools for automatic object recognition and detection, image enhancement and image transformation. In there, three novel borehole image log processing methods are presented. The first one is an automatic borehole breakout characterization method which uses transit–time image data logs (see Section 2 for further information) as source, and which is very useful to automatically extract geometrical information about these formations (such as breakouts area, volume or average orientation). The second one is a method for ultrasonic borehole image logs eccentricity correction based also on the data contained in the transit–time image log associated with it. The last one is an automatic pad removal filter for ultrasonic borehole image logs, which removes the artifacts caused by any wirelogging tools with pads that was introduced previously. Both these last two methods aim to enhance the contrast of the image, thus improving its quality and making it easier to be analyzed and characterized by geologists and petrophysicists.

The second field, which is studied in Section 4, Section 5 and Section 6, regards the use of state–of–the–art deep learning techniques for predicting petrophysical properties and geological information. We propose to use deep learning techniques to extract the most amount of information from electrical and acoustic images for both methods. The first method uses deep stacked autoencoders to predict permeability and porosity logs with an accuracy of 96% (in both cases). The second one uses deep convolutional neural networks to predict sedimentary facies with an overall accuracy greater than 97%.

In Section 2, we introduce the most common different types of data obtained from reservoirs, their impact on reservoir characterization and their main use in the oil/gas science field. In this Section the data available for this work is also introduced. In Section 3, some image processing methodologies are applied to borehole image data logs

as part of data preparation process that might precede the characterization step. Some of these methodologies are basic image processing tools, while others are novel methods that aim to improve the quality of borehole image data, such as automatic breakouts detection or automatic artifacts removal. Section 4 presents a brief introduction to deep learning techniques, emphasizing autoencoders and convolutional neural networks. In Section 5, deep convolutional networks are used on borehole image data logs for reservoir characterization and facies classification, as well as the results obtained for real reservoir data characterization. In Section 6, a novel methodology for petrophysical laboratory or wirelogs data prediction using deep learning techniques applied to borehole image data logs is presented, as well as the results obtained when applying it to real reservoir data. Finally, in Section 7, the conclusions regarding all the work are presented and discussed, as well as future work that might be related by this document.

Chapter 2

Reservoir data

As introduced in Section 1, obtaining a reliable model that might succeed in the task of reservoir characterization is extremely difficult. This task is essential because it is not only important to find out where the fuel reservoir is located, but also to know how the field surrounding it is, i.e., how it is composed and what are their geophysical properties, in order to be able to make predictions about how the soil might react to production and exploitation.

In their book *Geostatistical Reservoir Modeling*, (Pyrcz and Deutsch, 2014) describes porosity and permeability measures as *the main variables needed for reservoir characterization*. Most of the physical properties measured in reservoirs are usually considered because they are, at some level, correlated with porosity and/or permeability.

The importance of permeability and porosity measurements is linked to the fact that these two properties help us understand how the geological system that forms the reservoir is. These properties can be measured directly from core samples extracted from wellbores drilled on the reservoir. Even though these measurements are very expensive (due to the careful process by which they are obtained), they are still essential for reservoir characterization purposes. As obtaining core samples from the whole drilled interval would be extremely expensive, the total coverage of core samples extracted from the borehole is commonly low. This is the reason why the oil/gas industry has shown interest on indirect methods to estimate these measurements, like the one presented in Section 6, as an alternative to be able to extrapolate them to the whole wellbore. Before going into the different types of data on which we rely nowadays for reservoir characterization, it is essential to fully understand the concepts of porosity (total and effective) and permeability (which will be explained in the following Section).

There exist several different sources that are used either to map the reservoir volume or to extract physical properties from the reservoir rocks (Slatt, 2006), from which the main four of them are described in Table 1.

Table 1 – Summary of the main sources of data for reservoir characterization. Further information in Slatt (2006); Pyrcz and Deutsch (2014).

Data Source	Resolution	Coverage	Resulting data	Main application
Seismic	$\sim m$	$\sim km^2$	3-D volume	Subsurface Geological Reflectors Identification
Wireline	cm	–	Curve Logs	Reservoir Properties
Borehole Images	mm	–	Image Logs	Sedimentary structures[1]
Core Analysis	μm	$\sim cm^2$	3-D Scans/ Single-Measurement	Porosity/Permeability Measurements

2.1 Porosity and permeability measurements

2.1.1 Absolute porosity and effective porosity

Absolute porosity is defined as the fraction of void space contained inside a certain volume (Hook et al., 2003). Porosity values can range from 0%, when the volume is absolutely solid and has no inner void space, to 100%, when the internal void space occupies the whole volume[2]. In reality, porosity values for porous media analyzed in reservoir applications commonly go up to a maximum value of around 40%.

Nevertheless, reservoir experts are usually more interested on a different concept known as effective porosity. Effective porosity is defined as the fraction of interconnected pores contained inside the total volume of a certain porous medium. Non-interconnected pores, even though being void space inside the porous medium, are not considered in this calculation because they do not contribute to fluid flow through them and, therefore, they ought not be taken into consideration for permeability estimations.

In plain words, if we consider the porous medium as a solid with a volume V_t, which has a certain distribution of void structures inside, each one of them with a certain volume V_i, so that it satisfies

$$\sum_{i=1}^{N} V_i < V_t, \tag{2.1}$$

its absolute porosity (henceforth, PHIT) would be given by

$$PHIT = \frac{\sum_{i=1}^{N} V_i}{V_t}, \tag{2.2}$$

while its effective porosity (henceforth, PHIE) would be a value between 0 and PHIT. This concept is illustrated in Figure 2.1.

[1] One of the main objectives of this work is to prove that Borehole Image Data Logs can be used for Reservoir Characterization, as shown in Section 5 and Petrophysical parameter estimation, as shown in Section 6.

[2] It is obvious that a volume with a porosity value of 100% could never exist, as its own existence as a volume requires it not to be composed by absolute void. By definition a void volume is not a solid, but rather just void.

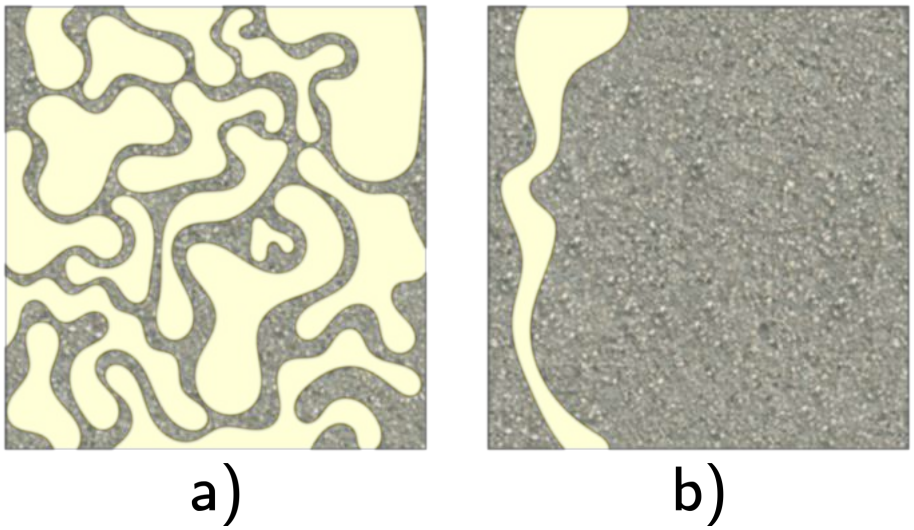


Figure 2.1 – Illustration of how the pore distribution affects the absolute porosity and effective porosity values. In this illustration pores are represented as clear yellow areas, while the rock matrix is represented in gray. As it can be seen, the volume in the left (a) has a much greater absolute porosity than the one on the right (b), however their pores are not connected and thus its effective porosity (pores that contribute to oil and gas production) is smaller than the effective porosity of the volume on the right (b). However, it is worth mentioning that in real conditions engineers can create artificial fractures in rocks like (a) in order to induce pore connection and thus, increase the effective porosity of the rock.

Pore interconnection is important in reservoir characterization because it describes the path that any flow could follow when going through that certain medium. Non-interconnected pores are void space inside the medium that cannot contribute to the flow of fluid, as there is no physical way the fluid can reach them. When core samples are available, calculating effective porosity can be achieved directly by using a porosimeter Storr (1988) or by computerized methods, like CT Scans Vega et al. (2014). However, extracting intact core samples is very expensive and technically difficult. For that reason, it is a common practice to estimate porosity values using indirect methodologies such as Wirelogs.

2.1.2 Permeability

The first notion and mathematical description of the phenomenon of fluid **absolute permeability** was proposed in 1856 by French engineer Henry Darcy on his book Darcy (1856). There Darcy described his study on the quantitative effect that different sand filters had on the output flow of water. Darcy noticed that the relation between the water flow and the applied pressure column, as well as the geometrical properties of the sand bed, required an extra proportional term K that proved to be constant for each kind of sand. Darcy called this property **permeability** and defined it as the physical property

that quantifies the ability that a porous medium has to allow flow through itself. The equation that links all these geometrical and physical properties is nowadays known as Darcy's law:

$$Q = \kappa \frac{A}{\mu} \left(\frac{\partial p}{\partial x} \right), \tag{2.3}$$

where Q is the flow of the fluid through the porous medium (in m^3/s), A is the cross-sectional area of the porous medium (in m^2), μ is the fluid viscosity (in $Pa \cdot s$), $\frac{\partial p}{\partial x}$ is the pressure differential through the medium length (in $\frac{Pa}{m}$) and K is the permeability of the medium (in m^2)[3]. The rocks found in a reservoir usually present permeabilities as low as $1\mu D$ for extremely low–permeable rocks, up to tens and even thousands of milidarcies for extremely good reservoirs.

Another measuring that is very much appreciated by Petrophysicist is the relative permeability per fluid for each sample. Relative permeability is defined as the ratio of the effective permeability of a certain phase to the absolute permeability of the sample, in multiphase flow in a certain porous medium Burdine et al. (1953); Honarpour et al. (1986). For oil industry purposes, this calculation is usually carried on considering two (oil and water) or three (oil, water and gas) different phases on each sample Stone (1973). Obtaining relative permeability measurements from core samples is a difficult and expensive task that provides a great amount of information about the properties of the reservoir rock and their dependance on the multiphase flow through them, unless artificial fractures are induced in the rock to cause pore connection.

Permeability measurements over the wellbore are basically use to feed dynamic-flow models that simulate the most likely reaction of that field to fuel production. In order to do so, it is usual to use permeability measurements to look for low-permeability layers (i.e., shales) which block the vertical flow of fluid through them.

Obtaining permeability values directly from a porous medium is not an easy task. It requires very specific and supervised laboratory tests, under very controlled conditions, to get reliable measures. This means that, up to the day this work is being published, there has not been invented any tool able to obtain permeability curve logs from reservoir boreholes. In reality, these measurements are usually obtained by two different mechanisms:

1. Direct laboratory measurements using lateral samples and core plugs.

2. Formation tests carried out in the reservoir, after or while drilling the wellbores, by which a certain (usually small) amount of hydrocarbon is extracted from within. By doing so, important information regarding how the rocks of the reservoir might react in real exploitation conditions can be obtained (Staff, 2006). Among all the

[3]Even though the international unit for permeability is m^2, milidarcies (mD) are widely used in several fields in science –i.e., in oil industry– for expressing permeability. One darcy is equivalent to $9.869233 \cdot 10^{-13} m^2$

information obtained by this method, it is possible to obtain an estimation of the permeability of the rocks present in the reservoir.

While the first method is very precise and provides the most reliable measurements, sampling the whole wellcore would be extremely expensive and almost impossible due to the fact that extracting this core intact is a very difficult task. On the other hand, formation testing is usually the final step right before the exploitation of the reservoir beggins, providing confirmation of all parameters and conditions concluded from the previous steps in the workflow (seismic geological layers delineation, well–to–well logs parameters correlation, core analysis, etc). Thus, formation tests can provide direct measurements of permeability (and several other parameters), but they are not used as direct sources to obtain them.

Apart from that, indirect estimations of permeability using theoretical models Timur et al. (1968) or using artificial intelligence and wirelogs (like the method explained in Section 6) are commonly used as an alternative to extrapolate and obtain approximate values of these parameters along different extensions of the reservoir. Although direct measurements are completely essential for reservoir characterization (and it is not the intention of this work to imply that indirect approximate measurements could ever replace them) it is important to notice that indirect measurements, even though less reliable, are much cheaper.

2.2 Seismic Data

Consider a seismic wave (mechanical wave) that is propagating through a certain medium (i.e., a certain layer on the earth). Its propagation velocity v_c depends on the acoustic impedance Z, as well as the density ρ of that specific medium (Doyen, 2007), and can be calculated as

$$Z = v_c \cdot \rho. \tag{2.4}$$

Now, when a seismic waves propagating at v_c encounters an interface between two different materials or layers, a fraction of it is bounced back by the boundary, while the rest continues propagating through the second material Allard and Atalla (2009). The fraction of the wave that bounces back is known as the reflected wave, while the part that continues propagating through the new layer is called transmitted wave. The amplitudes of the original wave (before the boundary) and the reflected wave are related by the *reflection coefficient* R, which depends on the difference of acoustic impedance of the mediums, and can be calculated as[4]

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1}, \tag{2.5}$$

[4]Equation 2.5 can be used only when assuming normal incidence. In real conditions, seismic waves encounter layer boundaries at different angles. For further reading on this subject see Allard and Atalla (2009).

where Z_1 and Z_2 are the impedances of the medium 1 and 2, respectively.

Seismology, which is the field of science that studies how mechanical waves travel and propagate through the earth, uses a modified version of the reflection relation presented in (2.5) in order to obtain valuable data from the Earth regarding the composition and disposition of its layers. In order to do so, engineers use a controlled seismic energy source (Hardage, 1985) (i.e., dynamite controlled blasts, specialized air guns or a seismic vibrator) to generate seismic waves in the surface, which propagate through the Earth and repeatedly bounce back when encounter any layer boundary. At the surface, special acoustic listening devices records the amplitude of the reflected waves, as well as the time that took for each response to reach the geomicrophones. By processing and stacking this data (Yilmaz, 2001) it is possible to obtain a representation of acoustic impedance local variations of the mapped Earth.

Seismic data is widely used nowadays in the oil/gas industry because it provides a great amount of information about the reservoir rocks (Chopra and Marfurt, 2007; Bath, 2013), the composition and the mechanical properties, such as acoustic impedance (which can be linked to density) of each one of the layers that compose the reservoir. In this scenario, reflection data has been used for reservoir characterization purposes such as lithofacies detection, rock properties extraction and petrophysical parameters mapping (Jiang and Spikes, 2016; Bosch et al., 2010; John et al., 2005; Mukerji et al., 2001), to name a few. An example of how reflectancy seismic data can be used for such purposes, after applying sequential inversion procedures (Lines et al., 1988), is shown in Figure 2.2.

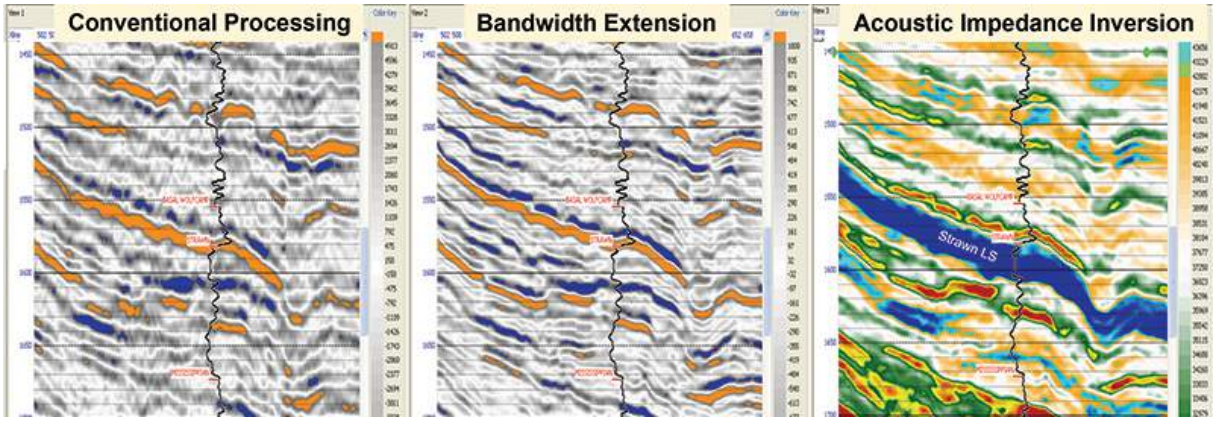
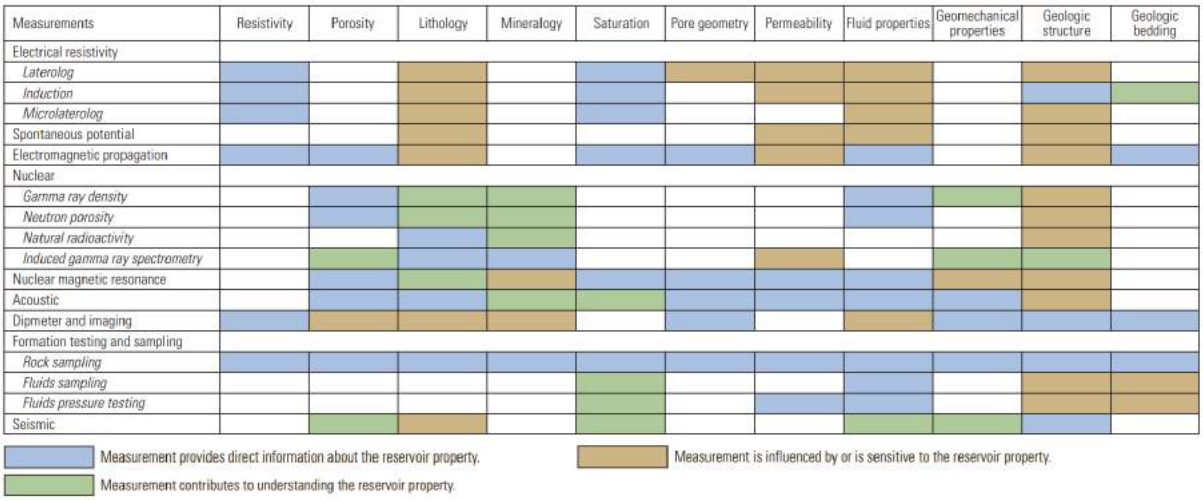


Figure 2.2 – Left and Middle: Example of reflectancy seismic data processed with two different methods. Right: Acoustic Impedance Inversion example obtained from the reflectancy data shown in the images on the left. The first two images (starting on the left) are maps of acoustic properties from a certain region of the earth. As it can be seen, it is possible to distinguish different strata that geologists usually link to geological layers (image on the right) by using data inversion methods. Source: DURHAM (2013)

Table 2 – Different types of wirelog tools and their relation to the physical property desired to be measured from the soil. Source: Schlumberger (2011).



2.3 Wireline Logs

As introduced in the previous Section, Seismic Data is widely used in the reservoir field to characterize the different layers of the earth that form the reservoir, as well as some basic mechanical properties such as acoustic impedance and rock density. However, these mechanical properties are usually not enough to characterize reliably the reservoir and therefore other types of physical properties of the rocks are required in order to fully understand the complexity of the field. With that purpose, the reservoir has to be drilled and opened so different tools can be lowered down by them and through the different layers of the Earth. A common borehole used for wireline logging can have more than one kilometer in depth, which means that it would be impractical, for economical reasons, to drill too many holes in the reservoir. Choosing the right spots that should be drilled in order to obtain the maximum amount of information about the soil while minimizing the effect caused on the reservoir, as well as the number of drills, is a difficult task that usually takes into account seismic data obtained previously Hughes (2002). In reality, the coverage area of the wireline logs data is much smaller to that of the seismic data, although its vertical resolution is far superior.

Each logging tool has the purpose of recording a certain physical property of the reservoir for each certain depth. Variations of these properties through the depth of the borehole usually indicate the variation of mediums or the presence of some structures that might be interesting to the experts. An example of four of these wireline logs is shown in Figure 2.3, where the distinction of layers containing hydrocarbons and brine is possible by analyzing the values of the Gamma Ray, Resistivity, Neutron Porosity and Bulk Density logs.

The main objective of wireline logging is to locate hydrocarbons in the field, as well as

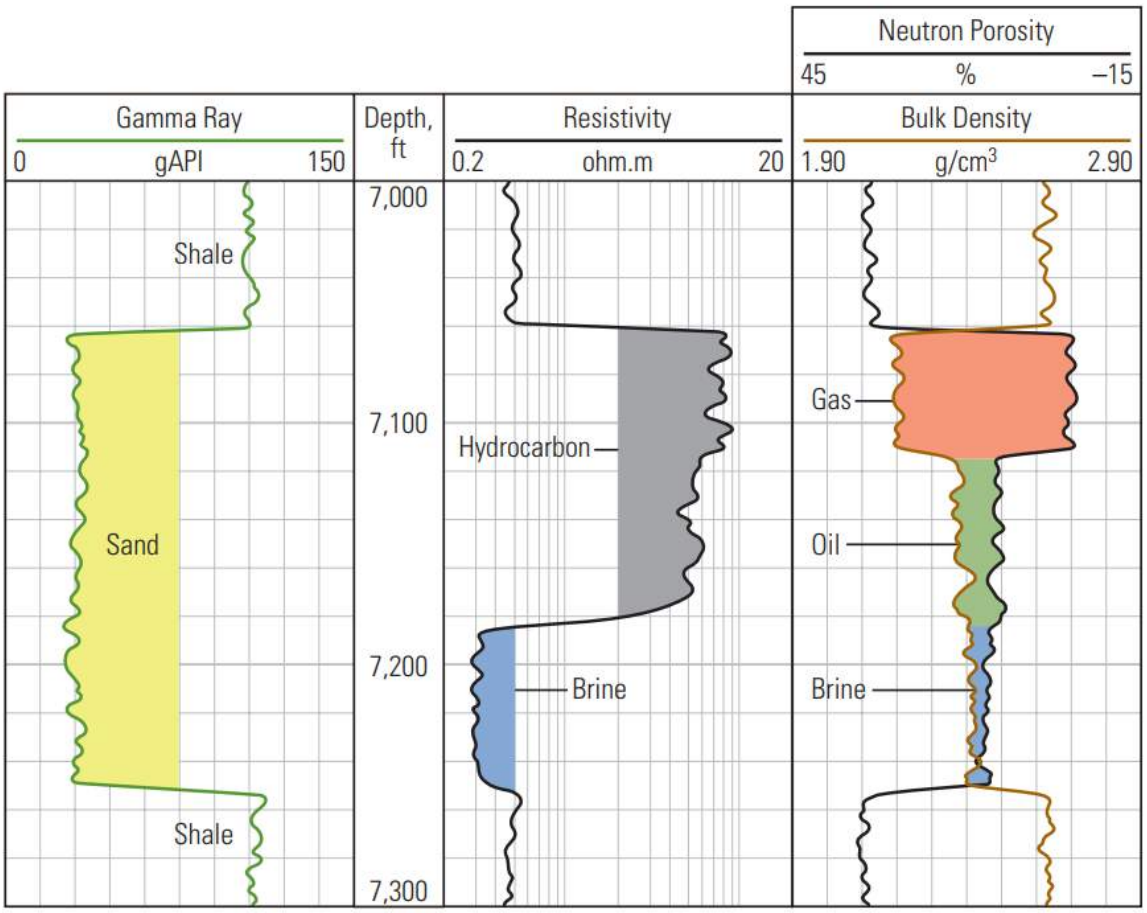


Figure 2.3 – Example of wireline logs: Gamma Ray, Resistivity, Bulk Density and Neutron Porosity. Source: Schlumberger (2011).

to obtain an estimation of the amount of it that it is present in the reservoir, so that the feasibility of field production can be calculated. This is the reason why most tools used in this operation measure directly or indirectly what is known as formation parameters: porosity, permeability and water saturation. With these parameters companies can obtain a fair estimation about the total amount of reserves contained inside the formation.

Interpreting wirelogs, however, is a complex task (Limited, 1984) that requires an experienced technician to analyze all logs side-by-side, along with seismic and core analysis data, altogether. A complete study on wire-logging tools, their principles and importance in reservoir characterization can be found in Ellis and Singer (2007).

2.3.1 Electrical Logging

Electrical wirelogs were the first wireline tools to be used in the field and nowadays the majority of wireline tools are based on this phenomenon. The principle behind this measure is that oil has a much higher resistivity than brine, and so by simply measuring the resistivity of the layers in the borehole it is possible to locate layers containing

hydrocarbons.

There exist basically three types of electrical logging tools which are:

1. **Laterolog** – The resistivity measurement is obtained by direct contact to the soil. The wireline tool contains a pair of buttons on its side that act as electrodes, where a certain potential is applied. This voltage creates a current that flows through the fluid contained in the medium that forms the borehole wall, thus allowing the measurement of resistivity. It is common to find two types of laterolog measures combined into a single logging tool called Dual Laterolog: Deep Laterolog (LLD) and Shallow Laterolog (LLS), which measure, respectively, the resistivity of the formation at deep and shallow depths into the layer. Laterolog measurements usually provide accurate measures for resistivity as low as 200 $m\Omega \cdot m$ and as high as 40 $k\Omega \cdot m$. Further information about this wirelog can be found in Sibbit et al. (1985); Chemali et al. (1988).

2. **Induction** – Some reservoir fields are composed by soils in which direct electrical measurements are not possible, i.e. non-conductive oil-based muds reservoirs or air-drilled boreholes. In order to be able to obtain resistivity measurements in these conditions an induction logging tool was created. This tool consists of an array of multiple transmitter coils driven by a certain alternating current. This current causes a magnetic field that induces continous circular eddy currents in the formation (Sophian et al., 2001). These currents, which are proportional to the formation conductivity, generate a secondary magnetic field which induces an alternating voltage in the receiver coils. Thus, the formation conductivity can be estimated by this method. Just like the laterolog tool, it is common to find Induction tools designed for different depths into the formation stacked together in the same tool (Induction Logging Medium (ILM) for medium depth into the formation and Induction Looging Deep (ILD) for deep depth into the formation) (Beste et al., 2000). At the same time, usually induction tools include a third measurement called Spontaneous Potential, which measures the spontaneous current created when each layer in the borehole is connected to a low-resistivity layer, i.e. in the surface of the soil, and which is useful to distinguish between shale and sand layers.

2.3.2 Nuclear/Radioactive Logging

The different materials that make up a reservoir, i.e. sand, clay, water and hydrocarbons, have very distinguishable chemical compositions. Some of these parameters, like the number of neutrons that the atoms of the material forming a certain layer has, can be directly related to petrophysical properties of interest to the petroleum industry, such as porosity or density. Thus, using tools able to obtain and map these chemical properties in a reservoir is a top priority, nowadays.

Nuclear/Radioactive Logging Tools base their measuring principle in the interaction between a certain type of radiation (usually nuclear radiation) and matter (reservoir layers) (Hertzog et al., 1989). When the radiation emitted by a radioactive source reaches matter different phenomena occur (Ellis and Singer, 2007). Part of this radiation might be initially absorbed by the material, energizing the electrons in the atoms and being emitted back later. Another part of it might be scattered a certain angle, while losing energy due to this effect. After the scattering process it is still possible that the remaining radiation keeps on going further into the material and is eventually absorbed by deeper layers. The radiation that is absorbed and emitted back by the material can be detected by a radioactive counting device set up in the logging tool. As the type of radiation used on these tools is highly energetic it can penetrate into much deeper layers than conventional logs (i.e., electrical logs), thus allowing us to obtain chemical information about the inner layers of soil.

There exist three major types of radioactive logging tool methods currently used in the industry:

1. **Bulk Density Log (RHOB)** – This tool sends Gamma-Ray radiation to the formation and detects the radiation that the formation itself emits back (due to the Compton Effect (Stuewer, 1975)). The amount of gamma-ray radiation that is absorbed by a given material present in the formation is directly proportional to the electron cloud surrounding its atoms[5], which is related to the bulk density of the formation (Blake, 1965). This method has been used for bulk density estimation for many years, and it is still essential for reservoir characterization purposes (Bertozzi et al., 1981).

2. **Neutron Porosity Log (NPHI)** – Neutron Porosity logging tools contain a source that produce neutrons. When these neutrons reach the formation, they get captured by it. Hydrogen atoms are the ones that have a bigger impact on capturing neutrons. Since this is the main compound that is contained inside the pores of porous reservoir formations (which is commonly filled with hydrocarbons or water), measuring the amount of neutrons that are absorbed by the formation along the depth of the well helps keeping track of its porosity (Tittman et al., 1966). The formation matrix, which corresponds to the main type of material that forms the analyzed layer, also has a certain impact on this measure, as absorbs a small fraction of the bombarded neutrons. That is the reason why RHOB measurements are usually calculated

[5]Notice that as gamma radiation consists on photons, they potentially interact with any kind of electrically charged particle. Nonetheless, the reason why nuclei is usually ignored in these types of gamma–ray & matter interaction is because the electrons in the nucleus are much more strongly bounded together with the neutrons than the electrons on the electron cloud of the atom are bounded with the nucleus. As the energy bound of the electrons in the electron cloud is far lower than the strong nuclear force present in the nucleus, most of the gamma radiation that interacts with the soil of the reservoir will not interact with the nucleus.

assuming three different matrices (Ellis, 1986): Limestone, Sandstone and Dolomite. It is the expert's task to use the right curve, for the right matrix, for each layer, which can be deduced by looking at different other wirelogs.

3. **Gamma Ray Log (GR)** – Unlike the rest of radioactive wirelog tools these tools do not emit any kind of radiation: instead, they only capture the radiation naturally emitted by the formation. There are basically three types of isotopes that naturally emit radiation in the Earth (Bigelow, 2002): ^{40}K , ^{232}Th and ^{238}U. These isotopes emit gamma-ray radiation in very distinguishable energy levels, which can be measured using a scintillation detector. Generally, shale layers have very small radioactive activity, while sands layers are usually detected using gamma ray logs because of the high values they cause in them. Recently, there has been used a new line of gamma ray logging tools which instead of single counting hits, obtain spectral readings, thus distinguishing not only the amount of radiation per depth, but also the energy of each type of radiation. These tools are called Spectral Gamma Ray Logging Tools. Counting the total level of the received energy distribution leads to the relative proportion of each one of the isotopes named previously in each layer of the soil. For further information about Spectral Gamma Ray Logs see Gadekea et al. (1991).

Other types of nuclear and radioactive logging tools less commonly used in the industry include Pulsed Neutron lifetime logs (Mills et al., 1988), Carbon Oxygen Logs (Lock et al., 1974) and Geochemical Logs (Herron and Herron, 1990). It is usual to find many radioactive tools stacked into a single logging tool.

2.3.3 Sonic Logging

The same concept applied for obtaining seismic data is also applied on borehole well-logging when using sonic logging tools (Cheng et al., 1992). In this case, pulses of pressure are applied to the well formation through the borehole mud by sonic transmitters. These pulses propagate through the formation in all directions. Far away from the transmitters locations some receivers can capture the propagating sonic wave. It is common to have several of these receivers before and after the transmitter, within the logging tool, so that undesirable effects, like noise, are minimized.

The average time that it takes for the wave to propagate through the formation, from the transmitter to the receiver, is called transit time, and it is usually expressed in μs. This transit time depends on two factors: the geological properties of the formation and the distance between the transmitter and the receivers. In order to discard the effect of the second component, the calculated transit time is divided by this distance, thus the sonic logs are usually expressed in $\mu s/ft$ or $\mu s/m$.

Sonic logs are usually correlated to the porosity of the formation, because the transit time of the sonic waves going through the formation depends on its physical properties and, therefore, on porosity and density. Thus, they are usually used along with radioactive logs to validate the porosity of the formation through the depth of the borehole. These logs have the advantage of not being much affected by the variation of size in the hole.

2.3.4 Nuclear Magnetic Resonance Logging

The phenomenon of nuclear magnetic resonance was described and measured for the first time by Rabi et al. (1991) and was, from that moment on, used in several fields like physics, chemistry, biology and medicine (Webb, 2013). The first wire-logging tool that applied the NMR principle for reservoir characterization was created in the 1960s by Chevron Research Laboratories (and others, 1964). Since then, the NMR Wireline Logging tool has been increasingly more important to geologists and petroleum engineers, as the data provided by it, as well as the different parameters that could be obtained from them, was better understood by science.

NMR technique measures the induced flux of hydrogen nuclei magnetic moment that form the tested sample. Within the oil/gas context, the fact that NMR only affects hydrogen nuclei (protons) means that NMR logging tools are almost exclusively affected by the fluids contained inside the pores of the formation, and not by the whole matrix (in opposition to what happens with sonic, electric and radioactive logs) (Kleinberg and Jackson, 2001). As the fluids inside the formation pores that are affected by NMR measurements are hydrocarbons (oil/gas) or water, this method is extremely useful for mapping their presence and volume over the borehole depth, which is the final goal of reservoir characterization.

From NMR logs geologist and petrophysicists might extract information regarding (Dunn et al., 2002), i.e., the volume and distribution of pore space (which is directly related to its absolute and effective porosity and can help predict permeability), the Water/Oil Proportion of the reservoir (which is extremely useful for reservoir production feasibility estimations) or the viscosity of the fluids contained inside the pores.

1. **Volume and distribution of pore space**: The response obtained by the NMR log tool captures the volume of fluid contained inside the pores of the formation. Therefore, keeping track of this volume over the depth allows geologists to know the total volume of pore space within the formation. This parameter can be linked to the formation porosity. On the other side, it is also possible to deduce the pore distribution which, along with a previous rock characterization, can give an estimation of the permeability of the formation.

2. **Water/Oil Proportion**: It is possible to infer the proportion of water and oil fluids inside the pore space due to the fact that the magnetic response of ^{1}H isotopes in

water is very different from the response of those isotopes in oil, and so they can be distinguished after processing the NMR data. This proportion is extremely useful for the reservoir production feasibility estimation.

3. **Viscosity Estimation**: One of the properties that shows correlation with the NMR responses is fluid viscosity. Thus, through data processing, it is possible to estimate the viscosity of the fluids contained inside the formation pores, which, in case the most part of the fluids in the tested samples are hydrocarbons, can be a reflection of the quality of the oil.

2.4 Borehole Image Logs

Borehole image logging tools allow us to obtain azimuthal high–resolution maps of physical properties of the wall of a certain borehole. These images, which have evolved into high-resolution data during the past decade (Morris et al., 2006), are commonly used for formation evaluation and characterization purposes such as fracture and breakouts identification, analysis of sedimentological features or evaluation and detection of bedding layers. An extensive study on this type of data and its applications can be found in Prensky (1999).

There are three different types of borehole imaging devices, which allow us to obtain four different types of borehole images: Optical imaging tools, ultrasonic imaging tools (which obtain both acoustic impedance and transit time images) and microresistivity imaging tools. Each one of these images is commonly used for a purpose, being the optical the imaging device that is less used nowadays. The following sections explain in detail each one of these data sources and their applications. Further information regarding borehole imaging tools can be found in Gaillot et al. (2007).

As the borehole imaging devices record the properties of the borehole wall, the final result can be considered as the inner peel of the borehole. Boreholes have cylindrical shapes, so the final images obtained must be flattened in order to be easily analyzed by the users. Therefore, one of the consequences of this procedure is that the apparent shape of the structures in the image are modified: i.e., a cutting bedding layer always intersects the borehole wall in a shape that forms a circle (or an ellipse if the borehole is not a perfect cylinder) which becomes a single-period sine wave when the borehole image is flattened (see Figure 2.4).

2.4.1 Optical Televiewers

Optical borehole imagers or televiewers were the first borehole imaging devices to be used for reservoir applications. This tool allows to obtain true-color images of the wall of the borehole, which can be then used by geologists to detect and identify layers

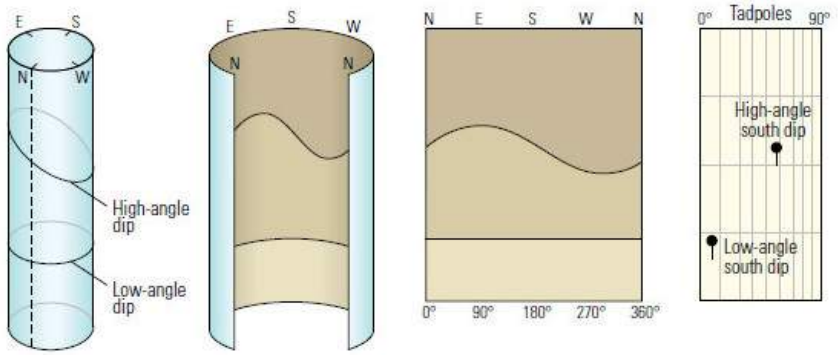


Figure 2.4 – Procedure of borehole image data flattening. As borehole data is obtained from a cylinder, it must be flattened in order to be displayed in a 2-D representation. This procedure changes the apparence of some structures, like the bedding fractures that appear in this schematic (here called high and low angle dips). Source: Schlumberger (2011).

and different lithological structures that might help them characterize the geology of the reservoir. One major drawback that this tool has is that they require that the borehole is filled with transparent fluid in order to be able to obtain acceptable data. In order to make this possible, these devices usually attach a fluid-injection system close to the optical lens. On the other side, it is also essential to include a lighting source to the device to overcome the darkness in the borehole.

Nowadays, optical televiewers are usually only used in air-filled boreholes, where the rest of televiewer types, i.e. acoustic and electrical, fail to obtain accurate images. Figure 2.5 shows an example of optical image data obtained from a real borehole.

2.4.2 Acoustic Televiewers

Borehole acoustic imaging logging tools, usually called acoustic televiewers, use ultrasonic pulsed waves to obtain information about the acoustic impedance of the borehole wall (Williamson et al., 1999). Differently than what happens with optical televiewers, acoustic televiewers can obtain valid data even with opaque muds in between the transducer and the borehole wall, which make them much more versatile than the optical imagers. Acoustic Televiewers are azimuthal imaging tools, which means that there is usually one sensor that rotates a certain number of times to cover all 360 degrees that form the borehole wall[6].

The measure principle on which these televiewers are based is the same used in the sonic logging and seismic tools: acoustic reflectivity. Figure 2.6 shows an schematic of the measuring process. In this case, the transducer emits a pulsed acoustic wave[7] that

[6]Most acoustic imaging tools have a resolution of 2 degrees per measure, which means that the final image has a total of 180 data points per row.

[7]The frequency of the acoustic pulse is directly related to the quality of the final image. Each manufacture optimizes its tools for a certain frequency, however it is common to find measurements in the frequency range $[250, 500]$ kHz.

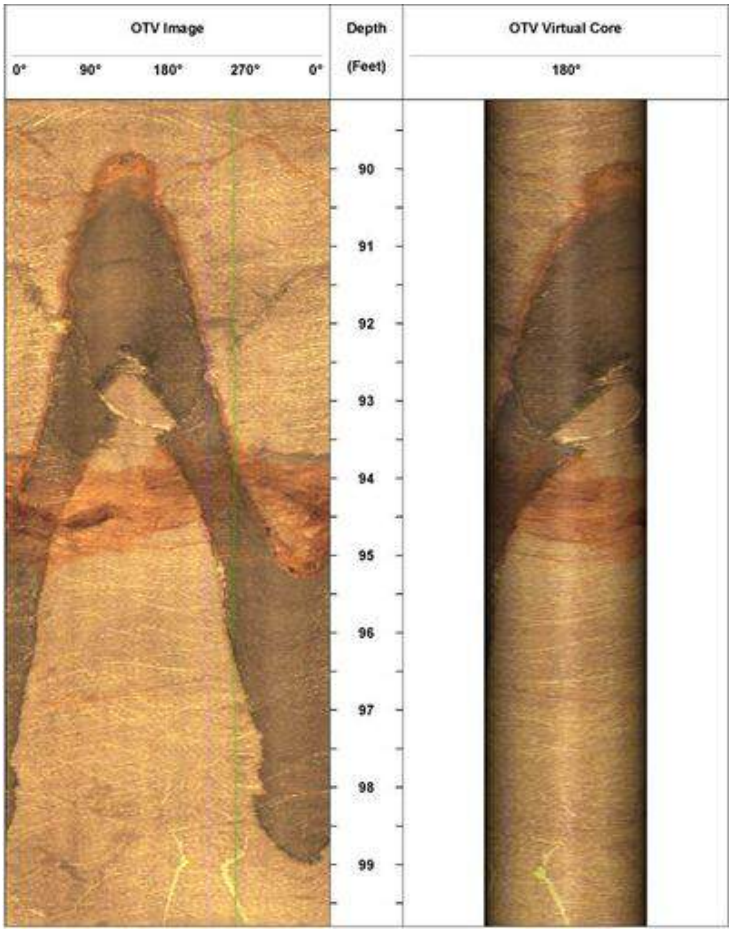


Figure 2.5 – Example of optical image obtained from a real borehole. On the left, the flattened true color image of the borehole wall. On the right, its 3D version (not-flattened). Source: Williams and Johnson (2000).

goes through the drilling mud and strikes the borehole wall. This first wave has an initial amplitude of A_0. While this wave travels through the fluid inbetween the logging tool and the wall, part of its power is absorbed by the fluid itself, according to a linear law that depends on the total travelled distance (the more distance, the more loss to the fluid). The total loss of amplitude due to the absortion of the fluid is refered to as ΔA [8], in this work. Once the wave reaches the wall, and as it was explained in Section 2.2, part of this wave will be absorbed by the formation and continue to propagate inside it, while the rest of it will bounce back. Being an azimuthal tool and assuming that the distance between the transducer and the borehole well is small, this reflection can be considered to happen under normal reflection conditions. In any case, acoustic televiewers extract two different values from each returning wave: the time that it took for that wave to reach the borehole wall and then reach the transducer back (transit time) and the ratio between the energy

[8]The drop of acoustic amplitude whilst travelling through the interface fluid depends not only on the distance between the probe and the borehole wall, but also on several other factors such as the density or the composition of this fluid. However, as modern drilling tools usually inject their own fluids to better control the drilling procedures, it is assumed that the only factor that effectively affects the contrast between pixels on the final image is the distance from the probe to the borehole wall.

of the emitted signal and the received signal (acoustic attenuation magnitude).

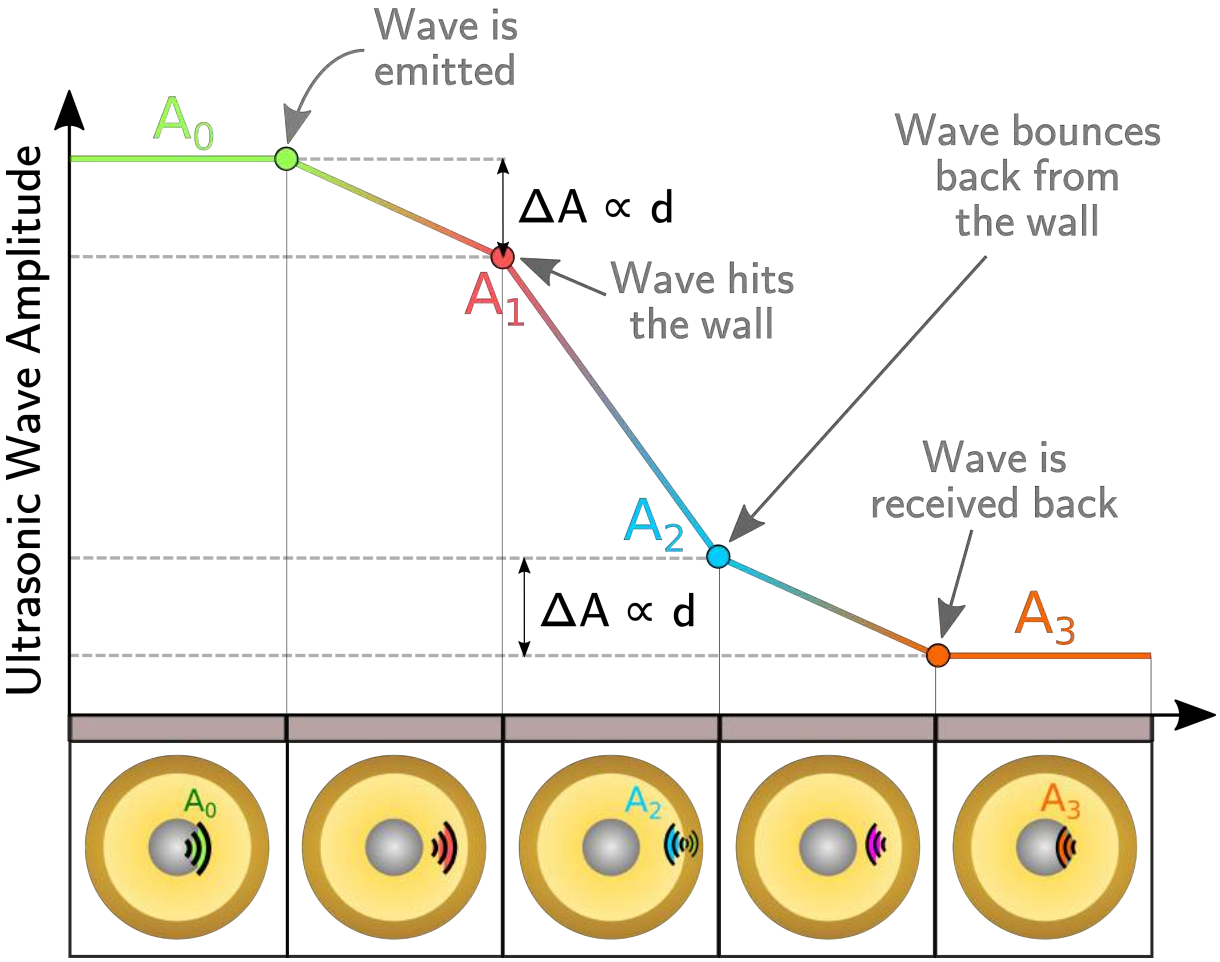


Figure 2.6 – Schematic explaining how the amplitude of the acoustic wave changes while travelling on the borehole: (0) First the sound wave is emitted with amplitude A_0; (1) while travelling through the drilling fluid (or the borehole water, in case no drilling fluid was used) the amplitude drops ΔA proportionally to the distance travelled by the wave, going from an amplitude of A_0 to A_1 right before the wave hits the wall of the borehole; (2) once the wave reaches the wall, part of it is absorbed by it and the rest bounces back to the transducer (the ratio between the absorbed and reflected part depends on the densities of the fluid and the wellbore rock) with an amplitude of A_2; (3) the reflected wave travels back to the transducer through the drilling fluid, losing again ΔA proportionally to the distance travelled by the wave, going from an amplitude of A_2 to A_3, right before the wave reaches the transducer back; (4) finally, the reflected sound wave reaches the transducer with an amplitude A_3, which is measured and stored as a pixel of the final ultrasonic borehole image log.

It is important to notice that the borehole shape is a very important parameter that is directly correlated with the final image quality. This is so because the drilling mud that separates the logging tool from the borehole wall absorbs a certain amount of the acoustic waves energy. This means that the more separated the transducer is from the borehole wall, the weaker will be the signal when reaching back the transducer. Variations in the

borehole shape affect the final acoustic image and causes regions with the same typical acoustic impedance to appear with different contrast and tone. Acoustic images obtained from elliptical and oval shaped boreholes suffer from artifacts caused by this effect. The most clear example of this is the difference in contrast caused by breakouts (for further detail see Section 3.3).

Transit time data is commonly used for characterizing the shape of the borehole. On the other side, Acoustic attenuation magnitude data can be used for lithology characterization, as it reflects the values of the acoustic impedance of the borehole wall. It is important to remember that the acoustic impedance of a medium is strictly related to its density. Thus, an image of the acoustic impedance of a borehole wall can be directly linked to a map of the different densities of the layers that form it.

An example of acoustic impedance and transit time data obtained from a well from the Santos Basin in Brazil is displayed in Figure 2.7 using the *Borehole Texture Analyzer* software designed in the context of this thesis.

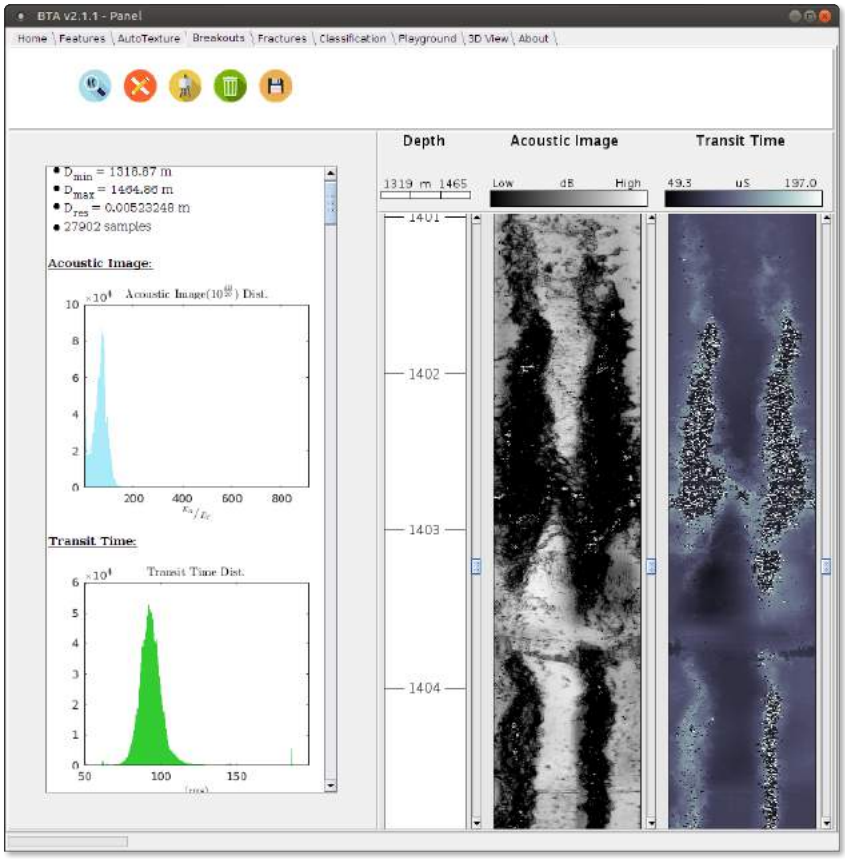


Figure 2.7 – Example of acoustic borehole log data (acoustic impedance and transit time) obtained from a well explored in the Santos basin in Brazil, displayed and processed using the software created for this work: Borehole Texture Analyzer (BTA).

Acoustic televiewers can be refered to using specific acronyms given by the manufacturer. Among them, the most usual are: UBI (ultrasonic borehole imager – Schlum-

berger), ABI (acoustic borehole imager – Advanced Logic Technology) and HRAT (high resolution acoustic televiewer – Robertson Geologging). For further information about Acoustic Televiewers, see Hayman et al. (1998).

2.4.3 Microresistivity Formation Imagers

Microresistivity formation imagers are logging tools that allow us to obtain azimuthal images of the microresistivity of the borehole wall with high resolution. In order to do so, this tool is equipped with a certain amount of electrode arrays, commonly called buttons, between which a constant potential is applied. Similarly to what happened in electrical logging tools measuring procedures (Section 2.3.1), this voltage causes a current to flow between the electrodes and through the material that forms the borehole wall. This current, which is measured by the device, is inversely proportional to the resistivity of the formation. As the buttons used in these tools are very small, to provide high-resolution data, electrical borehole imagers are usually called microresistivity formation imagers, as they map the microresistivity of the borehole wall.

In fact, these tools do not measure the real value of resistivity nor resistance of the borehole wall, but rather a value that is directly proportional to it. Thus, microresistivity formation images offer information regarding variations in electrical resistivity, which still helps geologists identify the different types of layers of the reservoir. The coverage of measurement obtained by these tools has evolved through the years, whilst the number of buttons on each array was increased. It is also worth mentioning that some of these tools require some type of fluid to be introduced between the electrodes and the borehole wall. It is usual to either perform an oil-based drilling procedure or use synthetic muds, which present just the right conductivity to allow resistivity measurements (Laastad et al., 2000).

Microresistivity images can be used directly for detecting structural features, such as fractures, faults or folds; sedimentary features, such as bedding surfaces, slumps or crossbedding; and diagenetic features such as stylolites (Bourke et al., 1989). On the other side, it is possible to use these images, along with some other data sources (i.e., wirelogs or acoustic image logs) to obtain further information regarding the borehole features, such as vugs, shale distributions, cementations, laminations, concretions and others. An example on using both acoustic and microresistivity image data for breakouts and fracture detection can be found in Tingay et al. (2008).

An example of microresistivity data obtained from a well from the Santos basin in Brazil is displayed in Figure 2.8 using the *Borehole Texture Analyzer* software designed in this work.

Electrical televiewers can be refered to using specific acronyms given by the manufacturer. Among them, the most usual are: FMI (formation micro imager – Schlumberger),

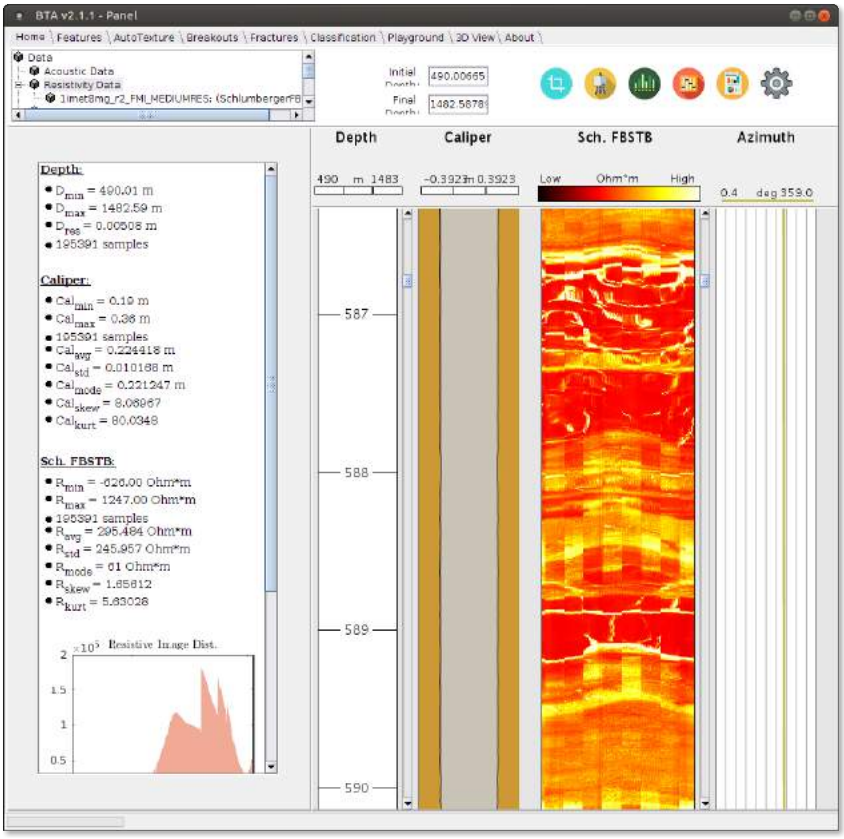


Figure 2.8 – Example of microresistivity image data obtained from a well explored in the Santos basin in Brazil, displayed and processed using the software created for this work: Borehole Texture Analyzer (BTA).

EMI (electrical micro imager – Halliburton) and FMS (formation micro scanner – Schlumberger).

2.5 Core analysis

Although the amount of information obtained using seismic, wireline logs and borehole imaging data sources is huge, it is usually required to extract raw samples either from the borehole core or the borehole wall in order to accomplish the fully complete reservoir characterization task.

Obtaining real samples from the reservoir offers the opportunity to validate the petrophysical properties estimated using the rest of the data, as well as to extract other types of properties, directly from the sampled rock, from direct evidences.

Core analysis can be defined as the direct study of samples obtained from the extracted cores of the borehole reservoir (Anderson, 1975; Craft and Keelan). The American Petroleum Institute explains in detail the recommended methodologies and procedures to be applied when dealing with core samples, as well as how to obtain petrophysical measurements from them, such as porosity and permeability calculations (API, 2010). It is possible to divide core analysis into three different subcategories:

1. **Plug analysis**: This is the most frequently used type of analysis. This procedure involves extracting a small sample of the whole core every certain distance (usually one every feet or so). Each one of these small samples is expected to represent a certain interval of the whole core. Intervals with high heterogeneity in lithology or any physical property (i.e., apparent pores grainsize) require more frequent sampling. Each sample is labeled with the depth of the core it was taken from and then several tests are run on them.

2. **Whole core analysis**: In some cases, the heterogeneity of the reservoir is so high that extracting plugs samples from the core would lead to unreal results about the reservoir lithology. In these cases it is possible to divide the whole core in smaller intervals and run tests on them.

3. **Sidewall core analysis**: There exist mechanisms to extract samples from the borehole wall, directly, while or after drilling it. These mechanism allow us to obtain direct samples from the reservoir. However, it has proved to be a very difficult task to keep these samples intact and undamaged, while retrieving them to the surface. This technique only provides unaltered cores when dealing with very specific types of formations (Craft and Keelan).

4. **Thin section granullometry**: This technique studies the grain size and sorting of thin sections of rocks extracted from the reservoir. The disposition of these grains has a great influence on the properties of the reservoir and the feasibility to exploit it, as it is intrinsically linked to permeability and effective porosity.

Regardless of the type of core analysis used, the physical properties that are most commonly extracted from them are (Keelan et al., 1972): Pore quantification and distribution, permeability calculation, fluid content determination, lithology characterization, dip analysis and density determination.

As it can be seen core analysis is essential to obtain properties that are extremely important for the oil/gas industry and that help enormously in the reservoir characterization process. Indeed, some of these petrophysical properties, like **Permeability**, can only be directly obtained by these means[9] or by formation tests, which make them even more valuable.

Despite their importance, obtaining intact core samples is neither easy nor cheap. In reality, it is much cheaper to obtain wirelog and borehole image data than a large number of core samples. This is one of the reasons why petrophysical parameters estimation from borehole logging data methodologies have evolved so much and so rapidly in the last years. These methodologies, like the ones discussed in this work, cannot only bring new information about reservoirs from different approaches, but can also help the oil/gas industry spare a considerable amount of resources.

For further detail into coring procedures and analysis methodologies see API (2010) or McPhee et al. (2015).

2.6 Available data

In this work, we use four different types of reservoir data: wireline logs, borehole images, petrophysical properties extracted from core analysis and lithology classification charts (made by petrophysicists). This data was provided by two different main sources: PETROBRÁS and ANP (National Petroleum Agency of Brazil, from the portuguese *Agência Nacional do Petróleo*). Table 3 shows the different type of data obtained from these sources, and how they have been used in our work.

All Image Data available for this work was obtained using borehole logging tools such as the ones described in Section 2.4. On the other hand, the available logs regarding petrophysical properties (PERM, PHIE and PHIT curves) were all obtained applying Timur-Coates models to NMR Data (Timur et al., 1968). Even though these measurements are not direct porosity/permeability measurements, it has been proved over the years that NMR extrapolated petrophysical data presents a high correlation with real

[9] Permeability can be estimated using different techniques and tools such as NMR Logs, however the only direct methods to obtain the true real permeability of a rock from a reservoir is to either handle a core plug obtained from it or to perform formation tests.

[10] PETROBRÁS

[11] This data cannot be published due to PETROBRÁS internal data security policy.

Table 3 – Summary of reservoir data available for this work.

Data Provider	Well Name	Basin (Location)	Wireline Logs	Borehole Images	Petrophysical Logs	Used in this thesis for
ANP	*1imet8MG*	17°25′23.0″S 44°38′27.5″W	GR,NPHI, PHIT,RHOB, LLD,LLS and others	UBI, FMI	n/a	Facies Classification, Well Synthesis
BR[10]	n/a[11]	n/a[11]	n/a	UBI, FMI	PERM PHIE, PHIT	Petrophysics Estim.

data (Mohaghegh et al., 1997). In any case, the data lent to us for this work had already been processed by this technique.

It is important to notice that all data obtained from PETROBRÁS was lend under their data secrecy policy, which means that it is strictly prohibited for us to publish any kind of reference to the location where this data was obtained, among other characteristics regarding the properties of the data. Contrarily, the data provided by ANP can be published as it was lent under research and scholarship terms.

Chapter 3

Image Processing applied to borehole image data logs

3.1 Introduction to digital image processing and applications

Digital Signal Processing (DSP) can be defined as the engineering field that deals with signals of different nature (electrical, computational, mechanical) and modifies them, by applying different methodologies and techniques, usually with the aim of extracting, enhancing or removing some sort of information from them (Antoniou, 2016; Smith et al., 1997). Even though it started as a field applied to electrical and electronics engineering exclusively, it rapidly evolved and gained ground with the evolution and expansion of computational technologies and has rightfully earned its position as one of the most important fields in science and engineering developed in the last century.

These days, it is very difficult to find a single field of science on which DSP does not play a crucial role: from astronomy to quantum physics or from chemistry to medicine, science nowadays relies on our capacity to acquire and process signals. It is important to clarify that although the term signal is wide enough to include images, they have become such an important source of information that the term DSP is almost exclusively used for 1–D signals that vary over a single or multiple variables (time, space, population age, etc.). Images are a special case of 2–D signals (or 3–D signals, when dealing with multiband images, i.e. color images) that represent the variation of a certain function over a spatial region. Images present spatial correlation, which means that each single measurement is conditioned by the values of its surroundings. For instance, color images represent the variation of light received by the capturing device[1] (i.e., a camera) in a certain region. 2–D representations of measurements that lack spatial correlation are usually called maps,

[1]Images can be obtained from different sources apart from the common visible–spectrum band, such as X-ray and CT-scan images, Gamma-Ray, UV and IR band images, thermal images, ultrasonic images, etc.

rather than images. The study of the digital techniques and methodologies that can be applied to images is known as Digital Image Processing (DIP).

The current state of the art of Digital Image Processing has changed and improved drastically over the past decade with the expansion of wavelets and multiscale analysis (Chui, 2016), as well as convolutional recognition and characterization techniques such as convolutional neural networks (LeCun et al., 1995) and deep learning techniques (LeCun et al., 2015; Schmidhuber, 2015). Apart from these, the most basic and common used image processing techniques (Gonzalez et al., 1992) can be summed up in:

1. **Gray–Level/RGB transformation**: Basic transformations in the levels of the image such as inversion (negative images), linear transformations (a.k.a normalization) and non–linear transformations (such as log, power–law or piecewise transformations). These transformations usually aim to change the values of the image for better visualization.

2. **Histogram processing**: Used to change the distribution of the intensity levels of the image so that its contrast is increased, thus increasing the level of visible details (e.g., histogram equalization, histogram matching or local enhancement).

3. **Image Restoration/Degradation**: Which seeks to remove a certain component of information contained inside the images, usually considered as noise, in order to restore an ideal image where only the desired part of the information is kept (e.g., band–reject or band–pass filters, notch filters, adaptative filters, etc.). These techniques rely on two different types of filtering processes:

    3.1. **Spatial filters**: Which work in the spatial domain (that is, with the original values of intensity of the image itself) and that are commonly used for smoothing, sharpening or averaging techniques (e.g., median, standard–deviation, gradient or canny filters).

    3.2. **Frequency–Domain filters**: Based on the fourier transform and the convolution theorem, which unlike spatial filters these rely on their fourier spectrum rather than on the images on the spatial domain themselves, and are extremely useful in the removal of periodic noise, for instance. Frequency filters are, without any doubt, the most important and widely used tools for DIP. The most common filters used in this methodology are: smoothing low–pass, butterworth or gaussian filters; sharpening high-pass, butterworth, gaussian and laplacian filters; and homomorphic filters.

4. **Image segmentation**: Used to partition the image intensity levels (gray–scale or color) into multiple groups that share a certain common property (such as similar color/gray–scale level or similar response to a certain filter). This technique is

widely used in different areas of knowledge as it allows further operations such as object recognition, morphology characterization or background removal. A special case of image segmentation is image binarization, which segments the image into two different groups.

5. **Image compression**: Which aims to compress as much information as possible from the original image by reducing their storage cost.

6. **Wavelets and multiscale processing**: Which seeks to characterize the input image, as well as extract and enhance several of its properties, by analyzing its response at different scales, thus widing the available information obtained from the image (Chui, 1992; Daubechies et al., 1991; Chui, 2016; Meyer, 1993).

As mentioned before, the most important tool used nowadays in image processing is the fourier transform. This tool allows us to represent any input signal as the series of periodic harmonic signals that compose it (Stein and Weiss, 2016). This is extremely useful because removing/enhancing a certain undesired/desired periodic component from a signal is much easier by using this technique than it would be by directly manipulating the original image in the original space.

The 2–D continous fourier transform ($\mathcal{F}$) of a 2–D signal input $f(x, y)$ (in our case, a grayscale image) is defined by

$$F(\omega_x, \omega_y) = \mathcal{F}[f(x, y)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot e^{-j(\omega_x x, \omega_y y)} dx dy, \tag{3.1}$$

where ω_x and ω_y represent frequency in both dimensions x and y respectively. On the other side, the 2–D discrete Fourier Transform of a discrete input $f(n_x, n_y)$ 2–D signal $F(k_x, k_y)$ is defined by

$$F(k_x, k_y) = \mathcal{F}[f(n_x, n_y)]_k = \sum_{k_y=-\infty}^{\infty} \sum_{k_x=-\infty}^{\infty} f(n_x, n_y) \cdot e^{-2\pi j \left(\frac{k_x n_x}{\lambda_{0_x}}, \frac{k_y n_y}{\lambda_{0_y}} \right)}, \tag{3.2}$$

where λ_0 represents the fundamental frequency of the signal (sampling rate) and k_x and k_y represent discrete frequency in both dimensions x and y respectively. Table 4 shows some of the most basic properties of the fourier transform. Among all these properties it could be said that the most fundamental one is Convolution, which allow us to apply filters in fourier space as a simple product of functions, rather than having to convolve them in the original space.

Table 4 – Basic properties of the fourier transform.

	Time Function	Fourier Equivalent
Linearity	$af_1(t) + bf_2(t)$	$aF_1(\omega) + bF_2(\omega)$
Time shift	$f(t - t_0)$	$F(\omega) \cdot e^{-j\omega t_0}$
Time scaling	$f(at)$	$\frac{1}{\lvert a\rvert} F\left(\frac{\omega}{a}\right)$
Time transformation	$f(at - t_0)$	$\frac{1}{\lvert a\rvert} F\left(\frac{\omega}{a}\right) \cdot e^{-\frac{j\omega t_0}{a}}$
Duality	$F(t)$	$2\pi \cdot f(-\omega)$
Frequency shift	$f(t) \cdot e^{j\omega_0 t}$	$F(\omega - \omega_0)$
Convolution	$f_1(t) * f_2(t)$	$F_1(\omega) \cdot F_2(\omega)$
	$f_1(t) \cdot f_2(t)$	$\frac{1}{\sqrt{2\pi}} \cdot F_1(\omega) * F_2(\omega)$
Differentiation	$\frac{\mathrm{d}^n f(t)}{\mathrm{d}t^n}$	$(j\omega)^n \cdot F(\omega)$
	$(-jt)^n \cdot f(t)$	$\frac{\mathrm{d}^n F(\omega)}{\mathrm{d}\omega^n}$
Integration	$\int_{-\infty}^{t} f(\tau)d\tau$	$\frac{1}{j\omega} \cdot F(\omega) + \pi F(0)\delta(\omega)$

3.2 Digital processing techniques applied to borehole image logs

The workflow used in oil industry for processing borehole images is well-stablished and has not suffered many changes in the last years, apart from constant upgrades and improvements regarding the techniques and methodologies.

The type of image processing method applied to borehole images depends basically on the type of data analyzed and the information that petrophysicists expect to get from them. In Section 2.4 the main uses for each type of borehole image data was introduced, as well as the main information to be extracted from them. The most typical workflows (Williamson et al., 1999) that both ultrasonic and microresistivity borehole images follow are shown in Table 5. Orientation and Speed Corrections are the most basic and initial operations carried out on borehole images. The first one is used to make sure that the image is oriented according to the true north azimuth, so that all information regarding orientation extracted from them is true (breakouts and fracture position and orientation, bedding plane orientation, etc.). On the other hand, speed correction ensures that the data displayed to the user has the right vertical size (due to the fact that the cable acceleration is not null and therefore some regions appear compressed while others are dilated).

After the data has been oriented and corrected, it is necessary to normalize it so it can be properly displayed to the user. Thus, Static and Dynamic enhanced data are obtained from the original data. It is worth mentioning that Static normalization do not perform any kind of operation in the histogram, but rather linearly scales the range of data to

Table 5 – Most common techniques and workflow that both Ultrasonic and Microformation borehole images follow to effectively extract information from them. The green tick (✓) symbolizes that the technique is commonly used for that type of data, while the red cross (✗) symbolizes that the technique is almost never used for that certain type of data. The yellow asterisk (❊) symbolizes that the technique is sometimes used for that certain data type, althought it might not be necessary.

Technique	Utility	UBI	FMS
Orientation correction	Used to correct the orientation of the data in order to be displayed with true azimuthal orientation	✓	✓
Speed correction	Used to correct the vertical position of each data line in order to be displayed with true depth correlation	✓	✓
Static normalization	Gray–Level normalization to improve visualization using the whole data	✓	✓
Dynamic normalization	Gray–Level/local contrast enhancement to improve visualization over a user–specified sliding window. This technique is useful to enhance local textural features, specially in FMS data (such as fractures or bedding planes)	❊	✓
Texture analysis	Histogram analysis that uses the proportions of gray–level values over a user–defined sliding window to characterize local differences in texture which might be related to some petrophysical or lithological property	✓	✓
Image segmentation	Used for porosity quantification, vuggy analysis (Cunningham et al., 2004) and breakouts detection	✓	✗
	Used for electrofacies characterization (Jungmann et al., 2011; Ye et al., 1998) and porosity evaluation	✗	✓
Edge detection	Used for fracture and bedding planes evaluation (Assous et al., 2013; Wang, 2005)	✗	✓

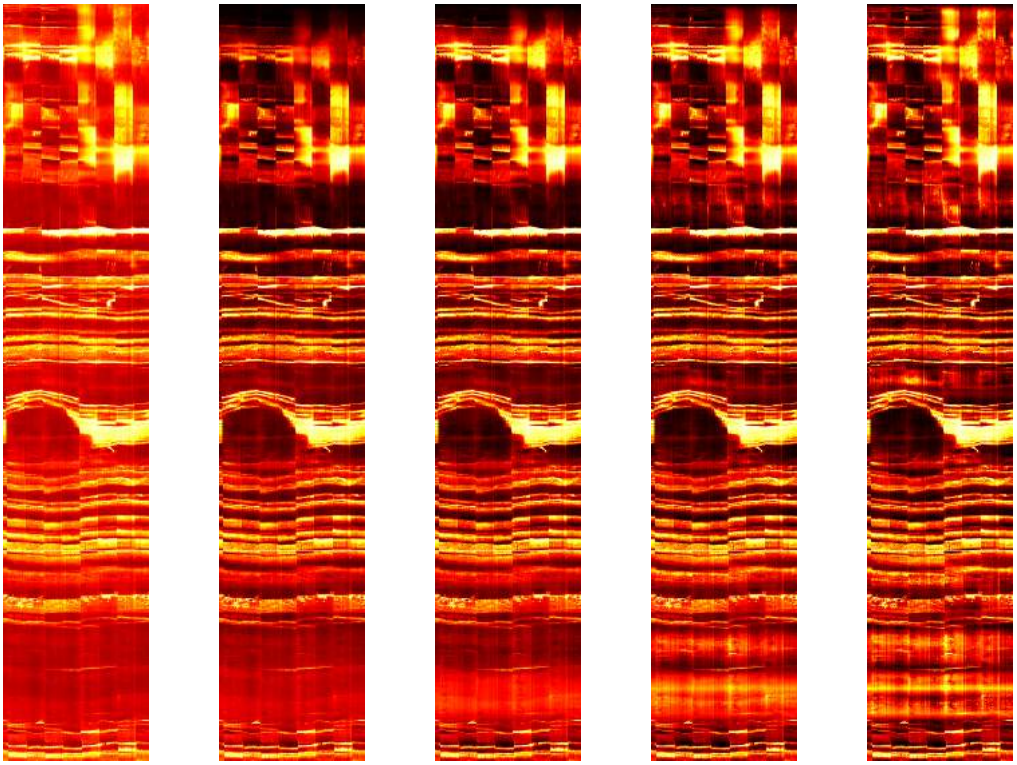
improve data display. Both these methods are based on a rather simple operation: for any input image A_{xy}, its static normalized image S_{xy} can be calculated as

$$S_{xy} = \frac{A_{xy} - \max_{i\exists[n]}(f_i)}{\max_{i\exists[n]}(f_i) - \min_{i\exists[n]}(f_i)}. \tag{3.3}$$

The difference between static and dynamic normalized images is that while static normalization uses the whole data to perform its operation, dynamic normalization uses a smaller sliding window, usually around 2 meters wide (but which can also be defined by the user), performing the normalization for each one of these windows before averaging

all results to form the final image. It is important to notice that features separated a great vertical distance from each other on dynamic images cannot be directly compared. Both these techniques compress the range of values in the data to $[0, 1]$.

Figure 3.1 shows the difference between static and dynamic normalization on a sample of microformation data: a) Shows the static data, while b) to e) show the dynamic data for a sliding window width of 1 m, 0.5 m, 0.25 m and 0.125 m respectively. As it can be seen the effect that the width of this window has on the final result is very noticeable. The smaller the width of the window, the more local features appear. Dynamic data normalized with small windows are usually used for fracture and bedding plane detection.



(a) (b) (c) (d) (e)

Figure 3.1 – *a) Static Normalized FMS Data; Dynamic Normalized FMS Data using sliding window of width of b)* 1 m *c)* 0.5 m *d)* 0.25 m *and e)* 0.125 m.

3.3 Automatic breakout detection and characterization from borehole image logs

As it was discussed in Section 2, the main objective pursued when drilling boreholes in the soil is to gather enough information about its geology, lithology and some of its physical properties in an attempt to help in the process of reservoir characterization. In order to obtain reliable data, the drilling procedure must be as noninvasive as possible, so that most of the original structures in the borehole wall are preserved once the logging

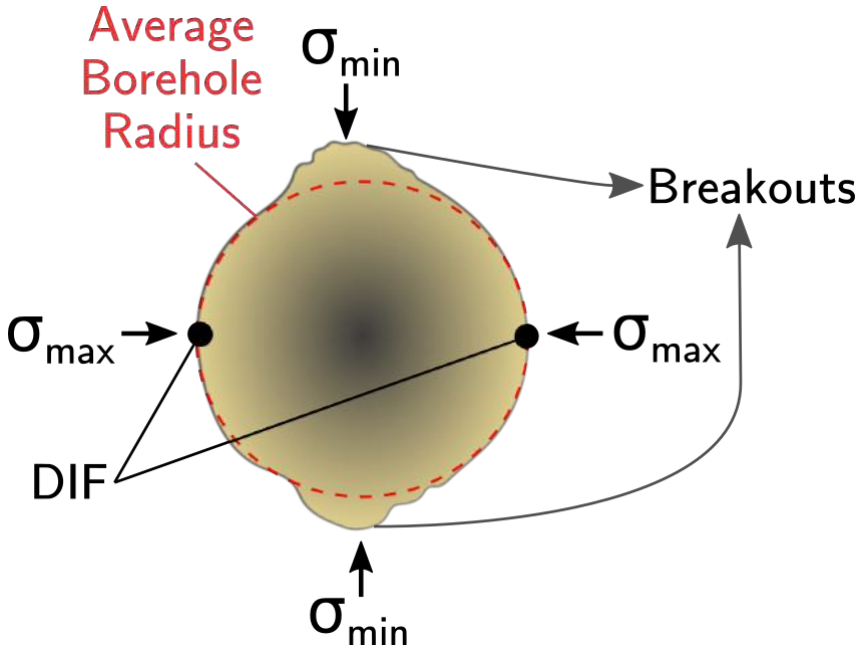


Figure 3.2 – Schematic showing how artificially induced fractures and breakouts are formed while drilling boreholes due to the imbalance of stresses. In this case a horizontal section of the borehole is shown. At this section it is supposed that the borehole has a nearly circular shape anywhere except in the breakouts regions, as it is represented by the average borehole radius circle. The regions where the drill applied the maximum pressure (σ_{max}) have greater chance to produce induced fractures (DIF), while breakouts only appear on the regions where the drill applied the minimum pressure (σ_{min}).

tools are introduced. Finding the balance between controlling the drilling pressure, speed or acceleration and doing this as fast as possible (so the process is not too expensive) is a very challenging task.

Most of the times artificially induced structures are created unintentionally (Kulander and Dean, 1988). The most common structures that are created due to mechanical stress while drilling are: drilled induced fractures (Aadnoy, 1990) (henceforth DIF) and breakouts (Haimson et al., 1986). Both these structures are formed when too much stress is introduced into the borehole. The stress applied by the drilling tool into the borehole can be decomposed into four main forces: two on each of the main axes orthogonal to the direction of drilling. Too much stress on the borehole wall might cause its rock to break, forming DIL, which appear in borehole image logs as vertical stripes that follow the maximum stress axis of drilling. On the other hand, this excess of stress might cause the part of the well with the less stress to collapse, leaving breakouts of material that appear as big dark marks on these logs. This phenomenon is illustrated in Figure 3.2.

It is important to notice that due to the intrinsical simetry of the borehole, breakouts almost always appear in pairs and separated 180° from each other[2]. Tracking the creation

[2]The only exception happens when one of the breakout–pair is still forming, so that only one of them appears on the final image. In any case, even geologists do not commonly agree amongst them on

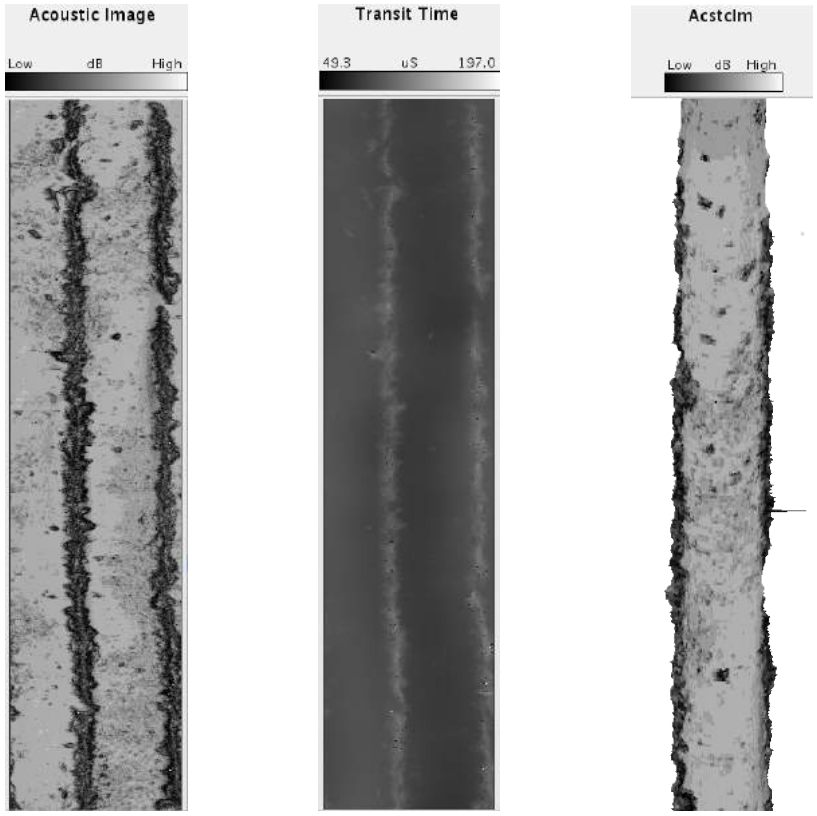
and evolution of these formations while drilling it totally essential, as an excessive amount of them will surely compromise the stability of the borehole and, in the worst case scenario, will make the borehole to collapse. The most common workflow during drilling is to keep track of several variables that help technicians and engineers to prevent the stress to grow excessively in the wellbore. Monitoring the formation of breakouts whilst drilling has gained popularity in the past decades, and is currently used as an alert monitor mechanism to control the mechanical stability of the borehole (Wessling et al., 2011).

Nevertheless, characterizing stress formations after the borehole has already been drilled has become more important over the years for evaluating wellbore stability (Barton et al., 1997) and in-situ stress determination (Barton et al., 1988). It is important to remind that these structures, even though they have little to do with the lithology of the reservoir, have a major impact on any further automatic textural/image processing technique applied to the borehole images obtained from the wellbore. For that reason, it is common to see experts detecting these structures manually, in a way to minimize the effect that they have on these tests. The use of these tools leads to very inaccurate measurements of their geometrical properties (i.e., area, volume, eccentricity or orientation), even in commercial softwares.

An example of what breakouts look like in different sources of ultrasonic borehole images is shown in Figure 3.3. As it can be seen in Figure 3.3a, breakouts appear as very dark predominant regions in this type of borehole data, not because the lithology at these regions is much different than the one from the surroundings of the breakout, but because the rock collapsing that formed the breakouts itself caused the wall of the borehole to be farther away from the transducer. As the drilling fluid between the wellbore and the transducer absorbs part of the energy of the sound wave emitted while travelling through it, the farther away from the sensor, the less energy the wave will have once arrives back to the transducer, and therefore the darker the region will appear in the borehole acoustic image (this will be further explained in Section 3.4.1). This is the reason why breakouts appear as brighter areas on transit time images, as it can be seen in Figure 3.3b: because in transit time images, brightness means farther away from the transducer, while darker means closer to it. A better way to visualize breakouts is using 3–D renderization, as shown in Figure 3.3c: here it can be seen that the breakouts do appear as prominent areas, changing the shape of the borehole to a more elliptical and asymetrical one.

Previous work on automatic detection of breakouts include the creation and use of a specialized tool to be introduced into the borehole separatedly (increasing the time that the drilling and acquiring operation would take, thus making this procedure more expensive) as proposed by Wessling and Pei (2015). Even though the main purpose of this method is to monitor the in–situ wellbore stability, it is worth noticing that obtains

whether these special cases should be considered breakouts. For that reason, we only consider breakout those formations that come in pairs and separated 180° one from each other.



(a) (b) (c)

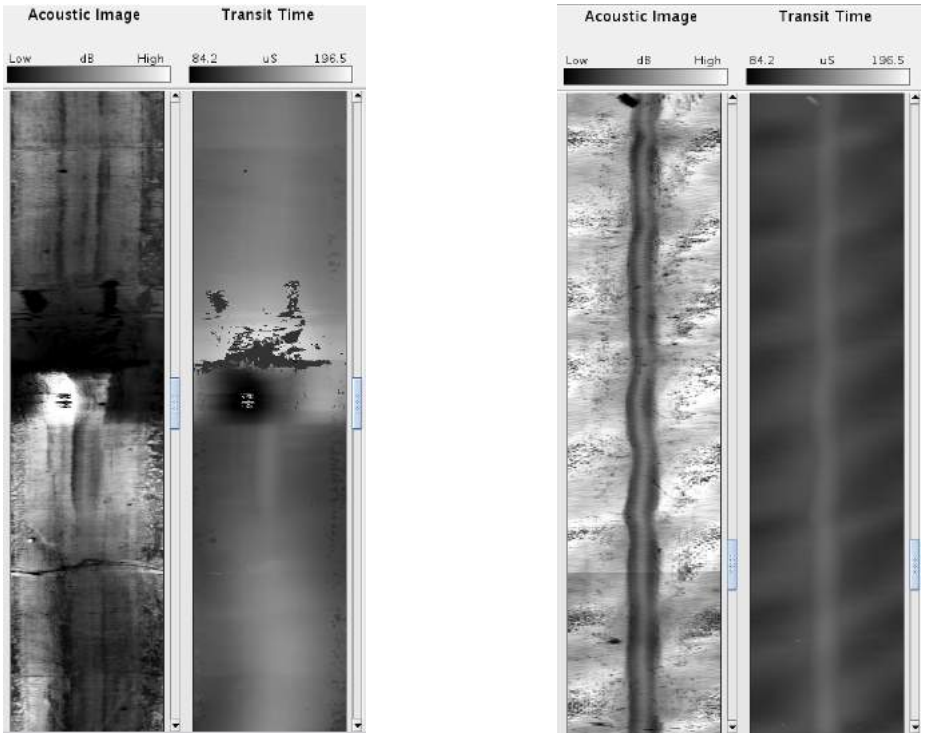
Figure 3.3 – Example of breakout structures in a real wellbore (*1imet8mg*, see 2.6) in a) an ultrasonic borehole image; b) transit time borehole image; c) 3–D view of the ultrasonic borehole image using the transit time data as the shape of the wellbore.

inaccurate results. In this Section, we present a novel technique that automatically detects and characterizes breakouts on borehole acoustic image data, very accurately. Our method uses both ultrasonic and transit time data as inputs, as well as several parameters that can be controlled by the user, in order to detect breakouts.

3.3.1 Proposed method

Even though breakouts appear as dark, separated regions in ultrasonic borehole images and very bright regions in transit time images, not all structures that match these conditions on borehole data are breakouts. In other words, there are different artifacts that might fullfil these conditions but that should never considered as breakouts. An example of such structures is shown in Figure 3.4. In Figure 3.4a, it is possible to see a structure, at the center of the image, that appears very dark in the ultrasonic data while very bright in the transit time image, even though it is not a breakout. The same happens in the vertical structure (denominated keyseat (Lofts and Bourke, 1999)) that is shown in Figure 3.4b.

Due to this kind of false–breakout structures, along with the fact that transit time



(a) (b)

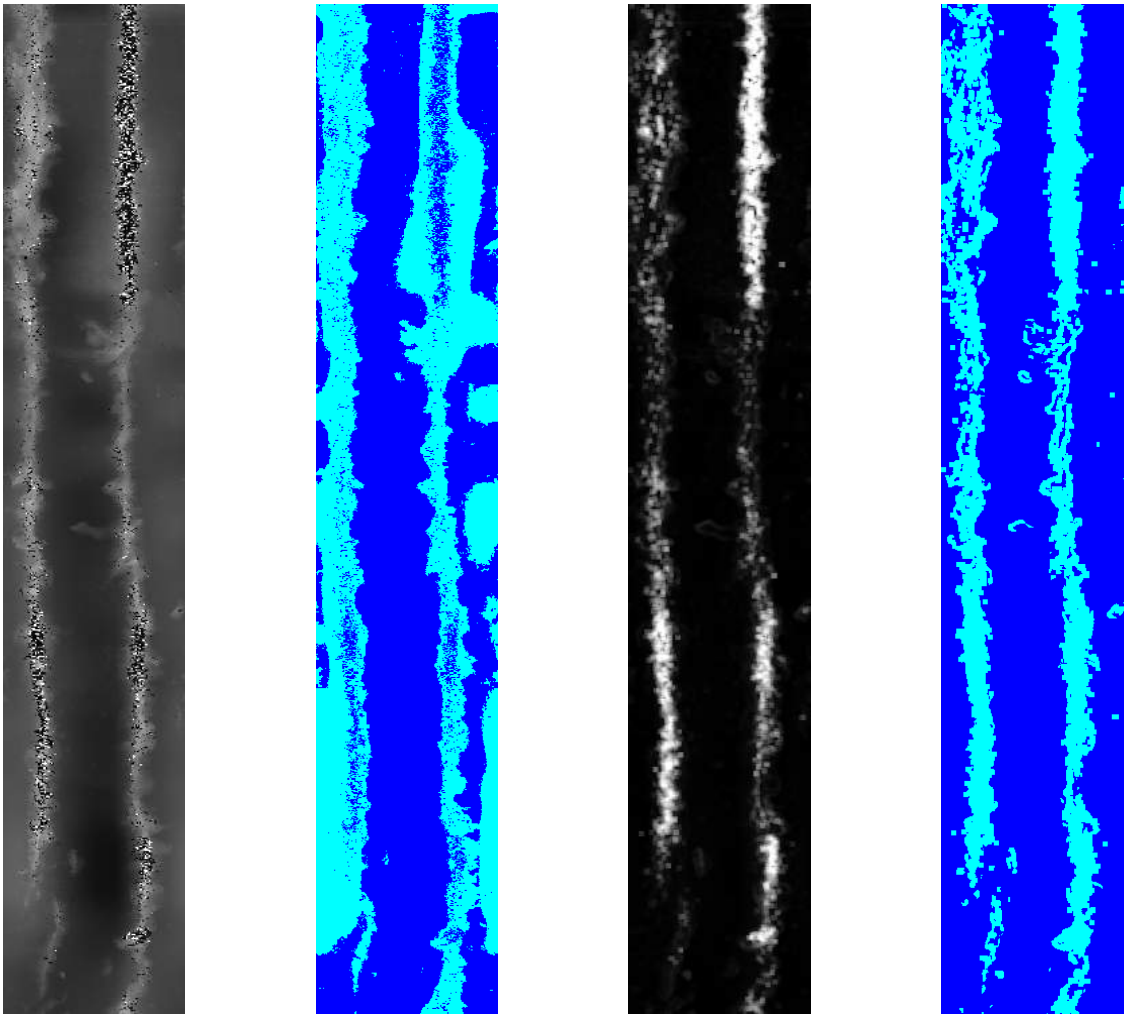
Figure 3.4 – Examples of two different wellbores on which some non–breakouts structures appear as dark regions on ultrasonic data and as bright regions on its equivalent transit time data. Neither the central region of image a) nor the vertical stripe shown on image b) are breakouts.

data is usually very noisy, the direct use of segmentation methods to ultrasonic and transit time data would most likely fail. Even though it would be expected for breakout regions to be homogeneous, in reality these regions present a high degree of noise, which affects directly the quality of the segmentation. Serendipitously, we discovered that breakouts regions can be characterized due to this heterogeneity and noise in the transit time data. Thus, we propose to first pre-process the transit time images with a standard deviation box-filter. This process enhances the regions with the higher level of local heterogeneity, which commonly happen to be breakouts.

The effect of this process to transit time data can be seen in Figure 3.5. First, in Figure 3.5a the transit time data is displayed. As it can be seen, the breakouts (both vertical formations that go through the whole image, separated 180°) appear as very bright and heterogeneous regions. If we tried to segment this data directly, as shown in Figure 3.5b, in an attempt to separate breakout formations from the background we would fail due to the intrinsic noise that these regions have. As it can be seen on that image, breakouts and non-breakouts regions are mixed, leading to a fail segmentation.

In Figure 3.5c the transit–time borehole data has been pre-processed using a standard deviation $[3 \times 3]$ box–filter before the segmentation process was applied. The final raw segmentation of this pre–processed data is shown in Figure 3.5d, and as it can be seen this

segmentation is far more accurate than the one achieved when using the raw Transit–Time data. Even though some of the breakouts regions were separated, this problem can be solved with further common post–segmentation processing (i.e., dilation, erosion, etc.).



(a) (b) (c) (d)

Figure 3.5 – Difference between the use of a) transit time borehole image data and b) its segmentation to characterize breakouts; and the use of c) transit time borehole image data filtered using a standard-deviation box filter and d) its segmentation. As it can be seen, the second segmentation, even when no further post-processes are applied, characterizes much more accurately the breakouts in the borehole data.

As introduced before, in order to obtain an accurate result it is necessary to apply some post–segmentation processing. First of all, and right after the standard deviation filtered transit time image has been binarized into breakout-candidate and background regions, a dilation is applied (Gonzalez et al., 2009). This procedure ensures that small regions that appear separated and unconnected after the binarization, which belong to the same breakout structure, are connected back. The size of the neighbourhood matrix on which the dilation depends is a parameter that allows the user to control this process.

Once the binarized image has been dilated, all detected structures (Breakout-candidate) are labeled. Thus, each structure can be distinguished from any other, which allows us to extract morphological information about them.

At this point, it is crucial to distinguish between those structures that strictly fullfil all characteristics to be considered as breakouts from those that do not. This can be achieved by comparing the morphological properties of each detected structure with those expected of a breakout pair. It is possible to identify accurately breakouts because these structures share the following morphological properties:

1. Each breakout formation must come in pairs. Breakouts are formed due to the axial stress on the borehole and therefore they are always formed in pairs. A single structure without its pair is not a breakout.

2. Breakout pairs must be separated 180°.

3. Breakouts can only appear with a certain range of orientations. They usually appear vertically but they might also be formed with a certain angle with respect to the vertical axis. It is impossible to find horizontal breakouts.

4. A breakout cannot be formed by more than two structures at each depth (see point one).

All these conditions are tested for each formation. Those that fullfil all of them can be considered as breakout candidates. Four examples of this method applied to real borehole data are shown in Figure 3.6, Figure 3.7, Figure 3.8 and Figure 3.9.

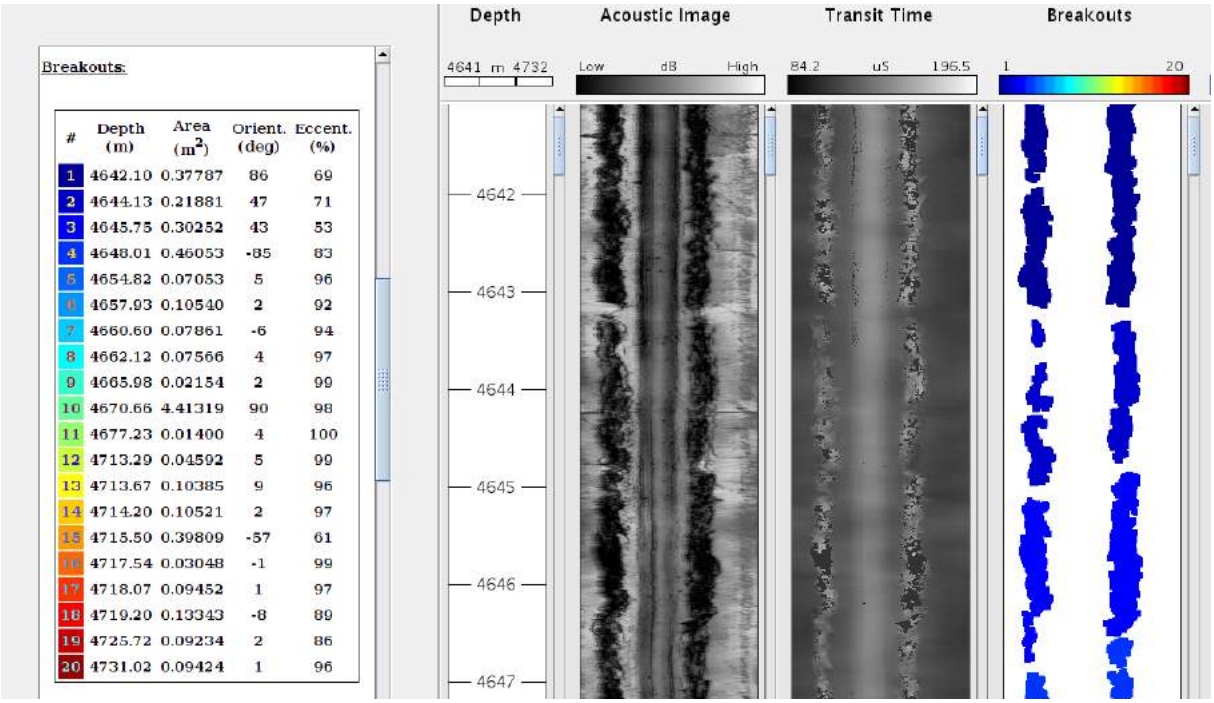


Figure 3.6 – Example of the method of automatic breakout detection and characterization proposed in this work applied to a real wellbore.

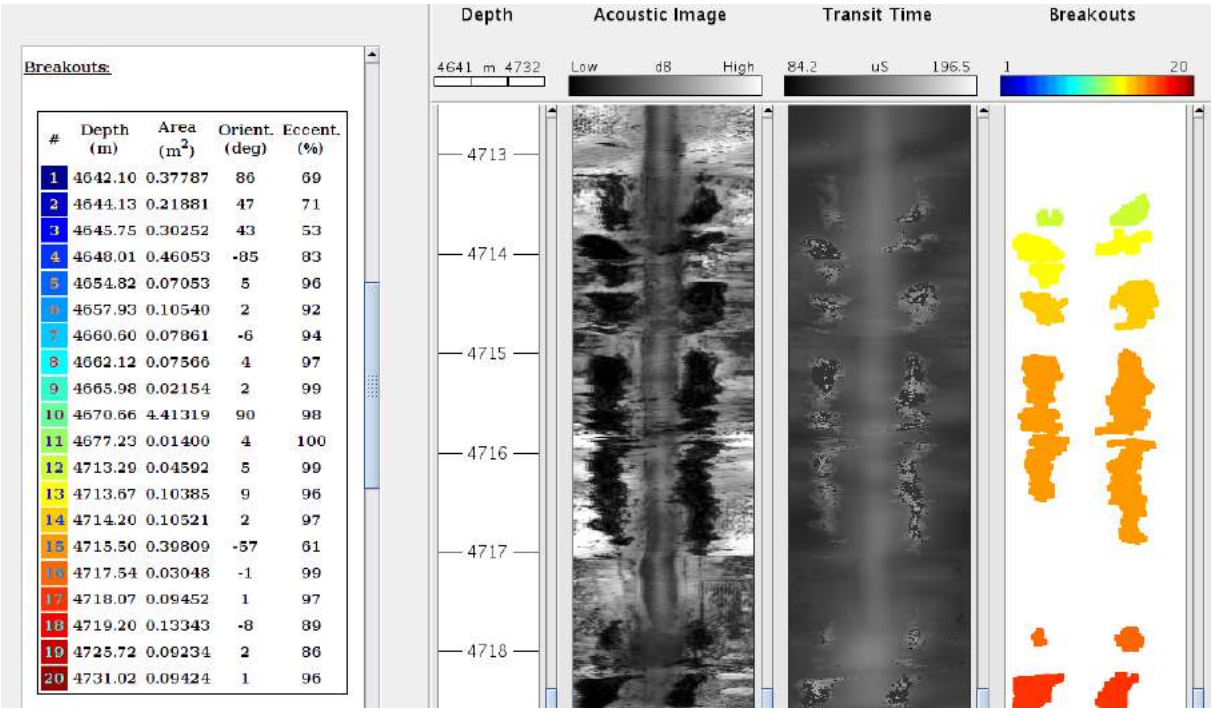
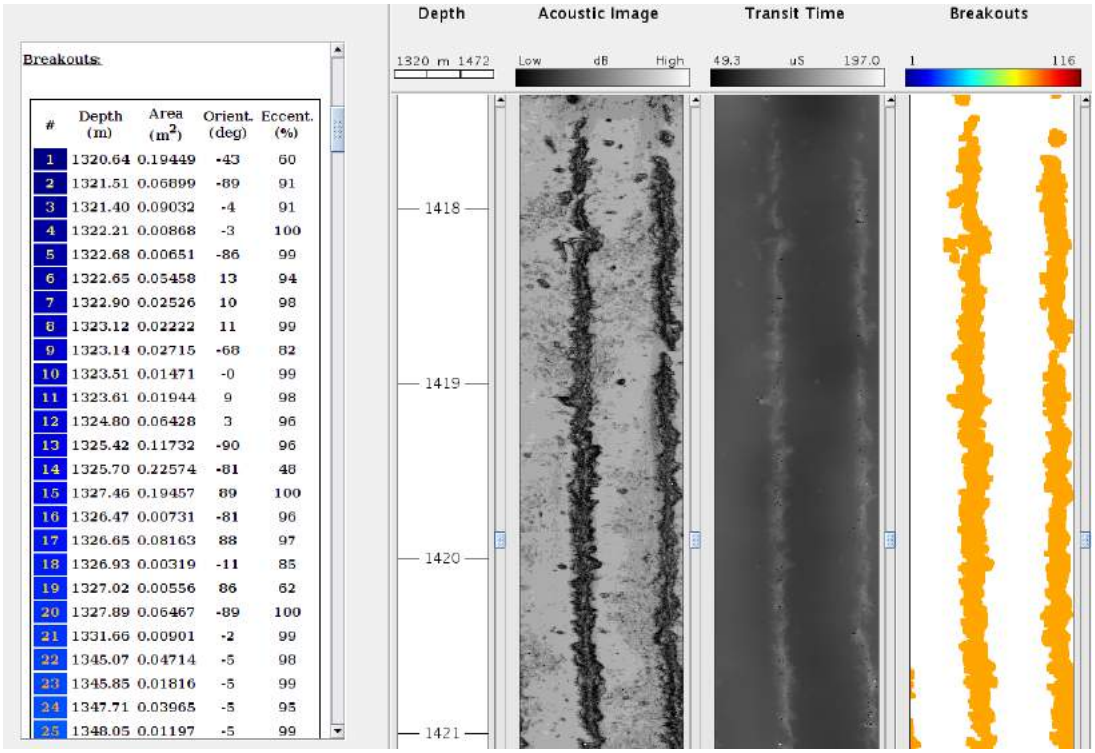
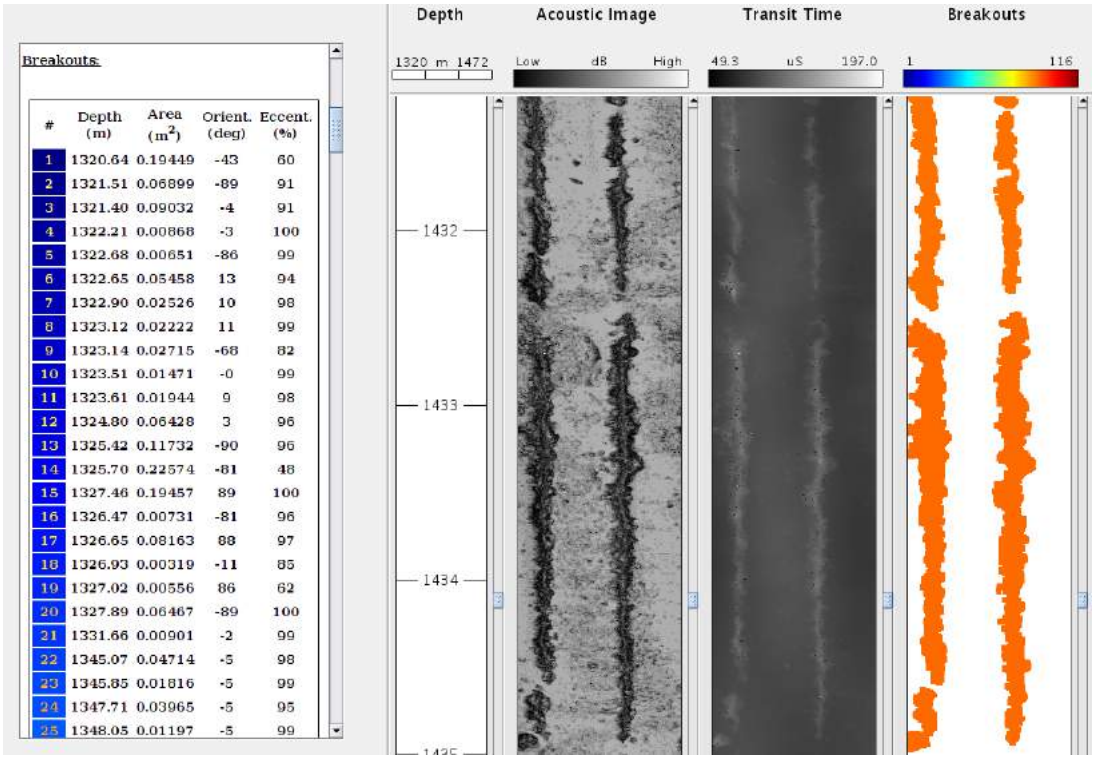


Figure 3.7 – Example of the method of automatic breakout detection and characterization proposed in this work applied to a real wellbore.



#	Depth (m)	Area (m^2)	Orient. (deg)	Eccent. (%)
1	1320.64	0.19449	-43	60
2	1321.51	0.06899	-89	91
3	1321.40	0.09032	-4	91
4	1322.21	0.00868	-3	100
5	1322.68	0.00651	-86	99
6	1322.65	0.05458	13	94
7	1322.90	0.02526	10	98
8	1323.12	0.02222	11	99
9	1323.14	0.02715	-68	82
10	1323.51	0.01471	-0	99
11	1323.61	0.01944	9	98
12	1324.80	0.06428	3	96
13	1325.42	0.11732	-90	96
14	1325.70	0.22574	-81	48
15	1327.46	0.19457	89	100
16	1326.47	0.00731	-81	96
17	1326.65	0.08163	88	97
18	1326.93	0.00319	-11	85
19	1327.02	0.00556	86	62
20	1327.89	0.06467	-89	100
21	1331.66	0.00901	-2	99
22	1345.07	0.04714	-5	98
23	1345.85	0.01816	-5	99
24	1347.71	0.03965	-5	95
25	1348.05	0.01197	-5	99

Figure 3.8 – Example of the method of automatic breakout detection and characterization proposed in this work applied to a real wellbore.



#	Depth (m)	Area (m^2)	Orient. (deg)	Eccent. (%)
1	1320.64	0.19449	-43	60
2	1321.51	0.06899	-89	91
3	1321.40	0.09032	-4	91
4	1322.21	0.00868	-3	100
5	1322.68	0.00651	-86	99
6	1322.65	0.05458	13	94
7	1322.90	0.02526	10	98
8	1323.12	0.02222	11	99
9	1323.14	0.02715	-68	82
10	1323.51	0.01471	-0	99
11	1323.61	0.01944	9	98
12	1324.80	0.06428	3	96
13	1325.42	0.11732	-90	96
14	1325.70	0.22574	-81	48
15	1327.46	0.19457	89	100
16	1326.47	0.00731	-81	96
17	1326.65	0.08163	88	97
18	1326.93	0.00319	-11	85
19	1327.02	0.00556	86	62
20	1327.89	0.06467	-89	100
21	1331.66	0.00901	-2	99
22	1345.07	0.04714	-5	98
23	1345.85	0.01816	-5	99
24	1347.71	0.03965	-5	95
25	1348.05	0.01197	-5	99

Figure 3.9 – Example of the method of automatic breakout detection and characterization proposed in this work applied to a real wellbore.

3.4 Automatic borehole image logs artifact removal

Borehole image logs have been used over the last decades for different purposes and are currently considered to be excellent tools that help greatly in the process of reservoir characterization, mainly because they can provide specific information about the field with a greater resolution than any of the rest of the tools used for reservoir characterization (wirelogs, seismic data, etc.).

Despite their importance, this data is commonly affected by some acquisition problems that might end up compromising its resolution and the quantity of information that can be effectively extracted from it. These problems, which are mainly mechanical or geometrical issues related to the drilling process, create artifacts in the borehole images that mask the information that really matters to petrophysicists. The main types of artifacts that affect acoustic and electrical borehole images (Al-Khabbaz et al., 2012) are:

1. **Drilling related artifacts**

    a) **Ghost hole**: In some reservoirs it is possible that two or more wellbores are opened at different locations in the surface but converge into a single hole further deep into the ground. In these cases holes are expected to appear close to the convergence point between the boreholes. It is important to identify these ghost holes that interconnect the wells before any processing is done to avoid critical errors such as porosity/permeability estimation errors. An example of the effect this artifact has on borehole ultrasonic images can be seen on Figure 3.10a[3].

    b) **Breakouts & DIFs**: As discussed in Section 3.3, too much pressure on the wellbore may cause its rock to break down and collapse, forming breakouts and DIFs. An example of DIFs on borehole ultrasonic images can be seen on Figure 3.10b[3].

    c) **Spiroloids**: Eccentricity and non-centered probes might cause this effect, which manifests itself as brighter marks that follow a helicoidal path though the wellbore (Sugiura, 2011). An example of the effect this artifact has on borehole ultrasonic images can be seen on Figure 3.10c[3].

    d) **Petal & Centerline fractures**: These artifacts are similar to DIF but instead of remaining parallel to the vertical axis, as DIF do, Petal fractures begin as vertical fractures in the edges of the wellbore and then change their orientation forming centerlines. They are caused by excessive pressure of the logging tool on mud-type fields. An example of the effect this artifact has on borehole ultrasonic images can be seen on Figure 3.10d[3].

[3]Source: Al-Khabbaz et al. (2012)

e) **Oil smearing**: Sometimes when the drilling process crosses the location of an oil reservoir part of this oil scapes from the reservoir into the wellbore and it is recorded into the image, appearing as vertical non-uniform stripes at random angular locations. This effect can also be produced if too much oil or drilling fluid is used while the drilling procedure. An example of the effect this artifact has on borehole microresistivity images can be seen on Figure 3.10e[3].

f) **Keyseat**: These artifacts are caused due to the contact between the drilling tool string and the wellbore wall. It can be clearly identified as a very sharp prominent vertical thick stripe that commonly appears on high–angle boreholes. An example of the effect of this artifact on borehole ultrasonic images was shown in Figure 3.4b.

2. **Logging related artifacts**:

a) **Pad pressure**: Some imaging tools (i.e., OBMI) require their pads to be in contact with the wellbore wall in order to obtain measurements. The pressure that the pads must apply on the wall, however, cannot be too high nor too low to avoid quality and resolution loss.

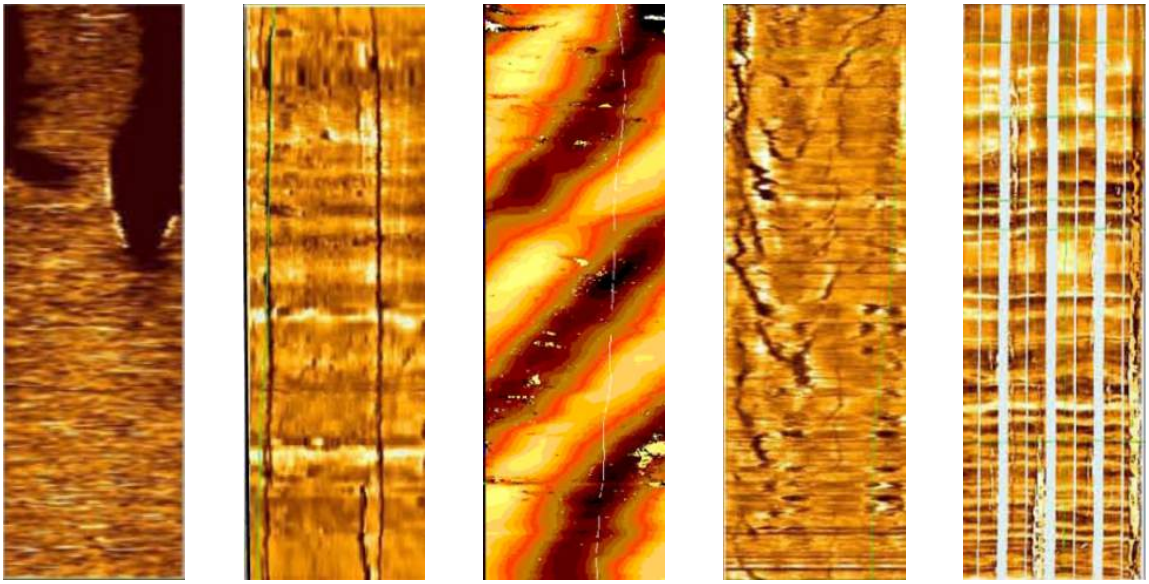
b) **Pad/Spring traces**: Imaging tools that have pads or strings (i.e., UBI, OBMI) usually leave their traces on some types of soft rock of the wellbore wall. These traces might be later captured by the tool itself, appearing on the images, interfering in a great manner in any further processing made on them.

On this chapter we present two techniques that allow us to deal with pad traces and uncentered probes related artifacts. In Section 3.4.1 we present an adaptation to Menger et al. (1994) technique, which uses travel transit time image data to correct eccentricity artifact on uncentered acoustic borehole images, increasing greatly the final contrast of the images. Additionally, in Section 3.4.2 we present a novel processing technique that removes the traces left by borehole imaging tool pads or strings, leaving only the background of the wellbore wall and thus increasing the amounf of effective information that can be extracted from the data.

3.4.1 Eccentricity Correction for Borehole Image Logs

As it was introduced in Section 2.4.2 acoustic televiewer images are based on measuring the variation of amplitude of the sound wave after being emitted from the transducer and right after being received back from bouncing back from the borehole wall. The physical principle behind this method is that the amplitude of the reflected wave depends on the ratio between the drilling fluid and the borehole rock densities, as expressed by

$$R = \frac{m \cdot \cos\alpha - \sqrt{n^2 + \sin\alpha^2}}{m \cdot \cos\alpha + \sqrt{n^2 + \sin\alpha^2}}, \tag{3.4}$$



(a) (b) (c) (d) (e)

Figure 3.10 – Examples of image artifacts on boreholes caused while drilling: a) ghost hole; b) drilling–induced fracture; c) spiroloid artifact; d) petal & centerline fractures; e) oil smearing.

where R is the acoustic reflectance coefficient of the interface between the borehole rock and the drilling fluid, $m = \frac{\rho_{rock}}{\rho_{fluid}}$ is the ratio between the density of the wall rock ρ_{rock} and the density of the drilling fluid ρ_{fluid}, $n = \frac{v_{fluid}}{v_{rock}}$ is the ratio between the velocity of the sound in the drilling fluid v_{fluid} and the velocity of the sound in the wall rock v_{rock}, and α is the angle on which the wave sound hits the borehole wall. Thus, the amplitude of the wave reflected by the borehole wall A_2 could be obtained by

$$A_2 = R \cdot A_1, \tag{3.5}$$

where R is the acoustic reflection coefficient of the interface medium and A_1 is the amplitude of the sound wave before hitting the borehole wall.

There is yet another effect that causes amplitude loss: the fluid travelling decay. While the sound wave is travelling through the drilling fluid –or the fluid found inbetween the transducer and the borehole wall, i.e., brine– its frontwave expands spherically. This expansion and the fact that the transducer has a limited size makes a certain part of the wave to be lost. The amount of amplitude to be lost is directly proportional to the distance travelled by the wave: the greater the distance, the greater the expansion of the wave and therefore the greater the loss of amplitude. This relation between *perceived* amplitude and travelled distance is usually approximated with an exponential decay, defined by

$$A_d = A_0 \cdot e^{-\lambda \cdot d}, \tag{3.6}$$

where A_0 is the original amplitude of the sound wave and A_d is the amplitude of the sound wave after travelling a certain distance d through a fluid with an acoustic decay

coefficient λ. If we assume that the interface fluid is homogeneous, according to the schematic presented in Figure 2.6 we could estimate the amplitude of the sound wave that would be received by the transducer A_3, after travelling a distance r_{θ_i} through the interface fluid with acoustic decay coefficient λ and being reflected by the borehole wall with a reflectance coefficient of R, and assuming that the original amplitude of the emitted wave is A_0, by applying the previous equations (3.4) and (3.6), as:

$$A_3 = A_0 \cdot R \cdot e^{-2 \cdot \lambda \cdot r_{\theta_i}}. \tag{3.7}$$

In reality the measurement recorded by the acoustic televiewer is not the final acoustic amplitude A_3 from (3.7), but rather the ratio between the amplitude of the emitted wave A_0 and the received wave A_3, in decibels (dB). This ratio Ω_{θ_i} can be obtained by applying (3.8). It is possible to deduce that, since A_3 must necessarily be inferior to A_0 and therefore Ω_{θ_i} must always be greater than 0. Ω_{θ_i} can be written as

$$\Omega_{\theta_i} = 20 \cdot \log\left(\frac{A_0}{A_3}\right) = 20 \cdot \log\left(\frac{A_0}{A_0 \cdot R \cdot e^{-2 \cdot \lambda \cdot r_{\theta_i}}}\right) = 20 \cdot \log\left(\frac{1}{R \cdot e^{-2 \cdot \lambda \cdot r_{\theta_i}}}\right). \tag{3.8}$$

Thus, it would be possible to correct the effect caused by the travelling decay by calculating a new ratio $\Omega_{\theta_i}\prime$, as shown in

$$\Omega_{\theta_i}\prime = 20 \cdot \log\left(10^{\frac{\Omega_{\theta_i}}{20}} \cdot e^{-2 \cdot \lambda \cdot r_{\theta_i}}\right), \tag{3.9}$$

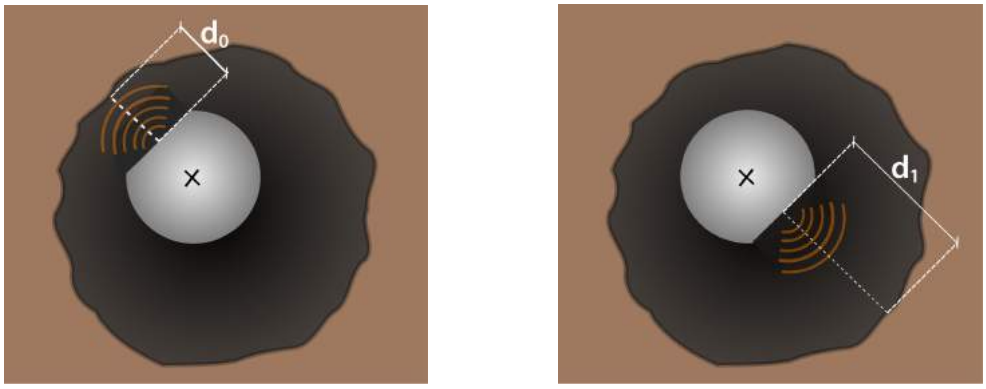
which can be simplified as

$$\Omega_{\theta_i}\prime = \Omega_{\theta_i} - 40 \cdot \log e \cdot \lambda \cdot r_{\theta_i}. \tag{3.10}$$

Borehole Imagers obtain one measurement per radial position (depth D_i and angle θ_i). Only if the borehole has the shape of a perfect cylinder and if the probe is perfectly centered at all times, will the distance between the transducer and the borehole wall be constant for all measurements, thus allowing us to ignore the effect of travelling decay[4]. In reality the shape of the borehole is very irregular, unsymmetrical and changes greatly at different depths, as it was shown in Figure 3.3c.

The irregular shape of the borehole and the uncentering of the probe make the distance between the transducer and the borehole wall to be different for each measurement. This effect is critical because it might cause two different areas of the borehole wall with similar rock properties to appear very different in contrast, which will clearly affect negatively the information extracted from this data. This is exemplified in Figure 3.11. Even though the real borehole distance (radius) image is not obtained while drilling nor measuring, Menger et al. (1994) presented a method to use transit–time logs to obtain an estimation to distance logs, which could be later used as an approximation of distance images.

[4]The variation of amplitude due to the travelling decay can be ignored if it is constant for all measurements only because images are usually normalized before displaying them. This normalization –or contrast adjustment– discards any constant component of the image, thus cancelling this underisable effect caused by the travelling decay.



(a) (b)

Figure 3.11 – Illustration of the effect of uncentered probes on borehole image logging. As it can be seen the transducer is much closer to the wall in a) than it is in b), which according to 3.6, will produce a difference in the acoustic impedance measurement, even if the rock in the wall at that depth is totally homogeneous.

In our work, we extend Menger's work by using transit–time image data instead of transit–time logs to correct Acoustic Borehole Image Logs. We know that the relation between the transit time at a certain radial position TT_{θ_i} (the time that it takes for a wave to travel all the way from the transducer to the borehole wall and arrive back to the transducer) and the real distance r_{θ_i} is given by

$$r_{\theta_i} = v_{fluid} \cdot \frac{TT_{\theta_i}}{2}, \tag{3.11}$$

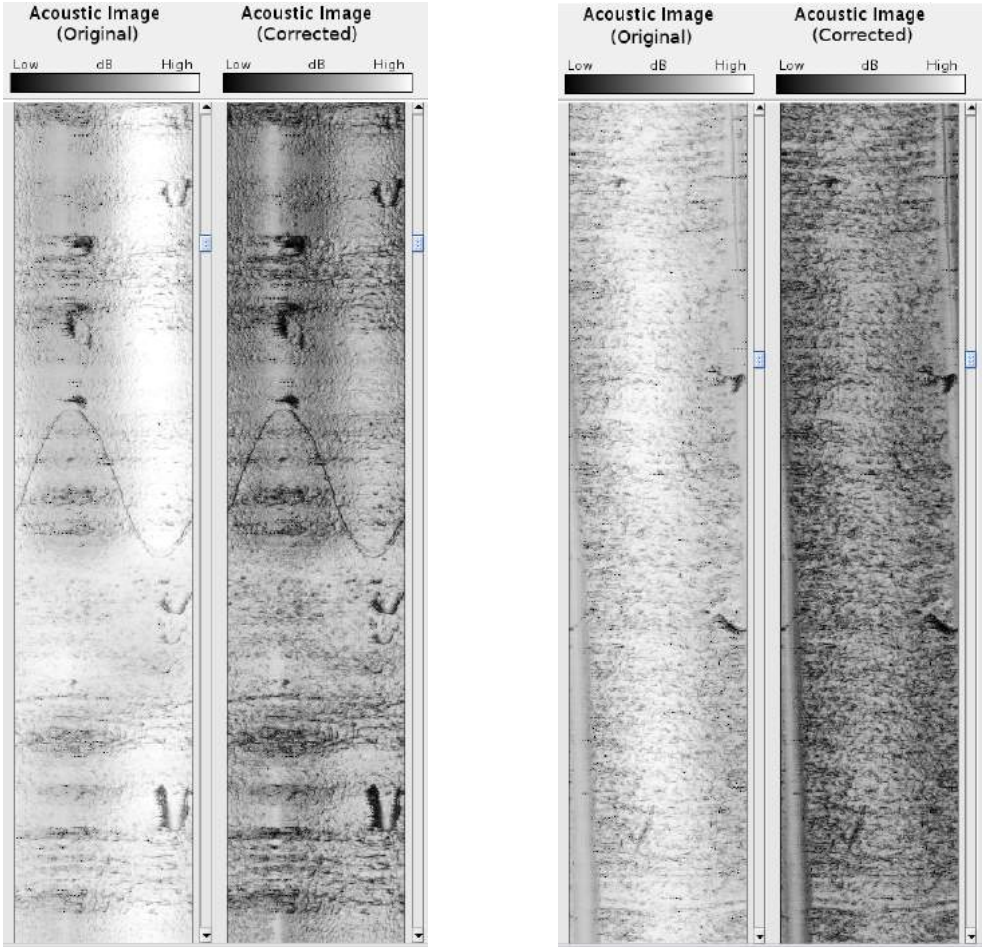
which we can combine with (3.10) to obtain

$$\Omega_{\theta_i}\prime = \Omega_{\theta_i} - 20 \cdot \log e \cdot \lambda \cdot v_{fluid} \cdot TT_{\theta_i}. \tag{3.12}$$

Most times neither the type of interface fluid nor the optimal λ coefficient are known by neither the user nor the processing software. Nevertheless, as in our work this tool has been implemented to obtain greater contrast in borehole images, so that further processing results can be more accurate, these coefficients can be set manually by the user. That way the user has a greater control on the enhancement process and is able to chose the coefficients ($v_{fluid} \cdot \lambda$) that produce the optimal enhancement for the post–process that is going to use that data.

Figure 3.12 shows an example of how this method increases the local contrast of the data. The images on the left on Figure 3.12a and Figure 3.12b show original acoustic borehole image logs from two different intervals that suffer from eccentricity and shape irregularities. As it can be seen in these two intervals, some parts of the borehole image appear to be much brighter than others (due to eccentricity while measuring). After applying equation 3.12 using a factor of $K = \log(e) \cdot \lambda \cdot v_{fluid} = 0.25$ the final enhanced images (on the right) show a much greater contrast than the original ones. In the enhanced

images it is possible to see details of the formation that were masked on the original raw images, ensuring that any post–process using the enhanced data will provide more accurate results regarding the real properties of the rocks in the borehole wall.



(a) (b)

Figure 3.12 – Illustration of the effect of uncentered probes on borehole image logging. As it can be seen the transducer is much closer to the wall in a) than it is in b), which according to 3.6, will produce a difference in the acoustic impedance measurement, even if the rock in the wall at that depth is totally homogeneous.

3.4.2 Logging tool pads removal

The presence of pad traces is one of the most common artifacts present on ultrasonic borehole images. These traces are left due to the mechanical system that the logging tool carries to ensure centralization and stabilization of the tool during the procedure. They usually appear as rectangular vertical stripes with homogeneous size and separation between them along the whole image, even though these parameters depend greatly on the logging tool used.

Removing these artifacts is an important procedure to be carried out due to the great impact that they might have on any further processing that might be applied to the images. In this section, we present a method that tries to model these traces in order to design a filter which, once applied to the original images, removes the pad artifacts, preserving all the lithological features present in the rocks of the borehole wall as if no pad traces had ever been there.

3.4.2.1 Pad–rejector filter design

If we assume that the pad traces are a multiplicative noise, then any borehole image u_{ij} on which a pad mark appears can be defined as the product of the original image q_{ij} (regarding the lithology of the borehole, the one we are trying to retrieve) and the noise added by the pads of the logging tool p_{ij} (which we are trying to remove), as

$$u_{ij} = q_{ij} \cdot p_{ij}. \tag{3.13}$$

Pad artifacts usually appear in borehole image logs as vertical rectangular stripes, with very sharp and well-defined edges, as it can be seen in Figure 3.13. Assuming that the shape of these marks is homogeneous over the whole borehole, we could try to model the traces that these pads leave p_{ij} on each line of u_{ij} as a square wave.

Square waves are defined by two parameters:

1. Period τ: in this case the space between each pad trace and the next one.
2. Duty cycle D: in this case the proportion between the width of the pad traces (λ) and the width of the gaps between them (μ).

If we define the width of our image u_{xy} as N, the number of pads as N_p, the width of the pad traces as λ and the width of the gaps between the pad traces as μ we could model p_{xy} as a vertical repetition of a square wave with a duty cycle D given by

$$D = \frac{\lambda}{\mu + \lambda}, \tag{3.14}$$

which period τ would be given by

$$\tau = \frac{N}{N_p}, \tag{3.15}$$

as it can be seen in Figure 3.13.

Instead of trying to find these pads by fitting or segmenting the original image, line by line, we will try to find the analytic expression of a filter h_{xy} that when applied to the original image u_{xy}, it removes or at least minimizes the effects of the pad traces. As we discussed in Section 3.1, filtering u_{xy} with h_{xy} is equivalent to finding the product of the fourier transform of $u_{xy} \xrightarrow{\mathcal{F}} U_\omega$ and the fourier transform of $h_{xy} \xrightarrow{\mathcal{F}} H_\omega$, as expressed in:

$$\mathcal{F}\left[u_{xy} * h_{xy}\right] = \mathcal{F}\left[u_{xy}\right] \cdot \mathcal{F}\left[h_{xy}\right]. \tag{3.16}$$

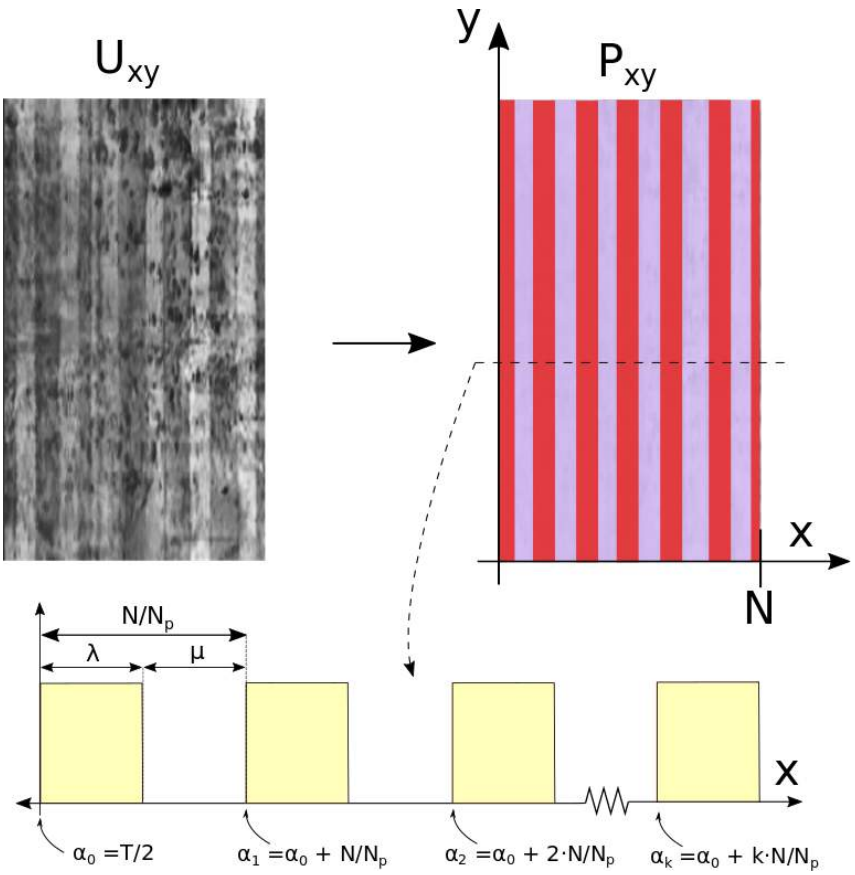


Figure 3.13 – Example of interval of acoustic borehole image log u_{xy} and its modeling p_{xy} using a vertically repeated square wave.

In order to do so first we need to obtain the expression, in space domain, that describes the square wave used to model the pad traces.

There exist different expressions that lead to square waves. Due to its periodic nature, it would be possible to use fourier analysis (Weller, 2015) to find the series of harmonics of periodic fundamental waves (sines and cosines) that, when weighted and added together, approximate a square wave, as expressed by

$$\hat{h_x} \approx \sum_{k=-\infty}^{\infty} A_k \cdot e^{jkx}, \tag{3.17}$$

where $\hat{h_x}$ is an approximation of the pad traces noise function we are aiming to define, A_k is the k$^{\text{th}}$ weight constant of each harmonic e^{jkx}, along the variable x. Even though this technique is widely used for similar purposes, it is well known that the number of harmonics required for a fair representation of a square wave with the least amount of noise possible is considerably high. A high number of harmonics would lead to complicate expressions, thus making the use of this method impractical. Another possibility could be to use the relation between sine or cosine waves and their absolute values, as expressed in:

$$\hat{h_x} \approx \frac{\sin \frac{2\pi x}{T}}{\left|\sin \frac{2\pi x}{T}\right|}, \tag{3.18}$$

where $\hat{h_x}$ is also an approximation of the sine–wave pad traces noise function we are aiming to define along the variable x and T is the wave period defined in (3.15). This last definition, however, might lead to complicate equations when applying the fourier transform due to the absolute value of the denominator.

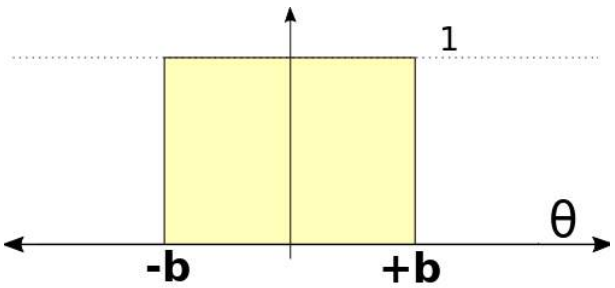


Figure 3.14 – Example of a pulse wave with its non–zero constant value between the interval $[-b, b]$.

Still, it is possible to define a square wave using the definition of a square pulse. A square pulse is a signal that has a non–zero constant value only during a certain interval, as shown in Figure 3.14. The analytical definition of a square pulse f_x defined on the interval $[-b, b]$ with an amplitude of 1 is

$$f_x = \begin{cases} 1 & -b \leq x \leq b \\ 0 & \text{otherwise} \end{cases}. \tag{3.19}$$

Using that definition it is possible to obtain the equation of a square wave g_x with duty cycle D as as an infinite summatory of square pulses shifted a certain number k of periods (considering that the period of that wave would be $T = \frac{2b}{D}$), as:

$$g_{xy} = \sum_{k=-\infty}^{\infty} f\left(x - \frac{2b \cdot k}{D}, y\right). \tag{3.20}$$

By applying the definition of the continuous fourier transform introduced in (3.1) to (3.19) we obtain the fourier transform of a single square pulse, written as

$$F_{\omega_x} = \mathcal{F}\left[f_x\right] = \frac{1}{\sqrt{2\pi}} \int_{-b}^{b} e^{-j\omega_x x} dx = \frac{1}{\sqrt{2\pi}} \frac{\left[e^{j\omega_x b} - e^{-j\omega_x b}\right]}{j\omega_x}. \tag{3.21}$$

It is posible to apply Euler's formula (Euler et al., 1748, Chapter 8)

$$\begin{aligned} \sin x &= \tfrac{e^{j\theta} - e^{-j\theta}}{2j} \\ \cos x &= \tfrac{e^{j\theta} + e^{-j\theta}}{2} \end{aligned} \tag{3.22}$$

to (3.21) and obtain

$$F_{\omega_x} = \mathcal{F}\left[f_x\right] = \sqrt{\frac{2}{\pi}} \frac{\sin \omega_x b}{\omega_x}. \tag{3.23}$$

We can now obtain the expression of a periodic square wave as an infinite sum of shifted square pulses. Thus, applying the time/space shift and the linearity properties (see Table

4) to equation (3.20) and taking into account the definition of F_{ω_x} expressed in (3.23), we obtain

$$G_{\omega_x} = \sum_{k=-\infty}^{\infty} \sqrt{\frac{2}{\pi}} \frac{\sin \omega_x b}{\omega_x} \cdot e^{-j\omega_x \cdot \frac{2bk}{D}}. \tag{3.24}$$

Since our image has a limited size we do not require a full square wave but rather N_p periods of it. Therefore, using the previous definitions, as also presented in Figure 3.13, we can obtain the final definition of N_p periods of a square wave H_{ω_x} with a duty cycle of D and a period of $\frac{N}{N_p}$, as a model of the pad traces left by a logging tool with N_p pads on an image with a width of N pixels, as expressed by

$$H_{\omega_x} = \sqrt{\frac{2}{\pi}} \frac{\sin\left(\omega_x D \frac{N}{N_p}\right)}{\omega_x} \cdot \sum_{k=0}^{N_p-1} e^{-j\omega_x \frac{N}{N_p} \cdot \left(\frac{D}{2}+k\right)}. \tag{3.25}$$

In Figure 3.15 the magnitude (on the top) and the phase (on the bottom) of the filter expressed in (3.25) when $D = 0.5$, $N = 180$ and $N_p = 6$, is shown.

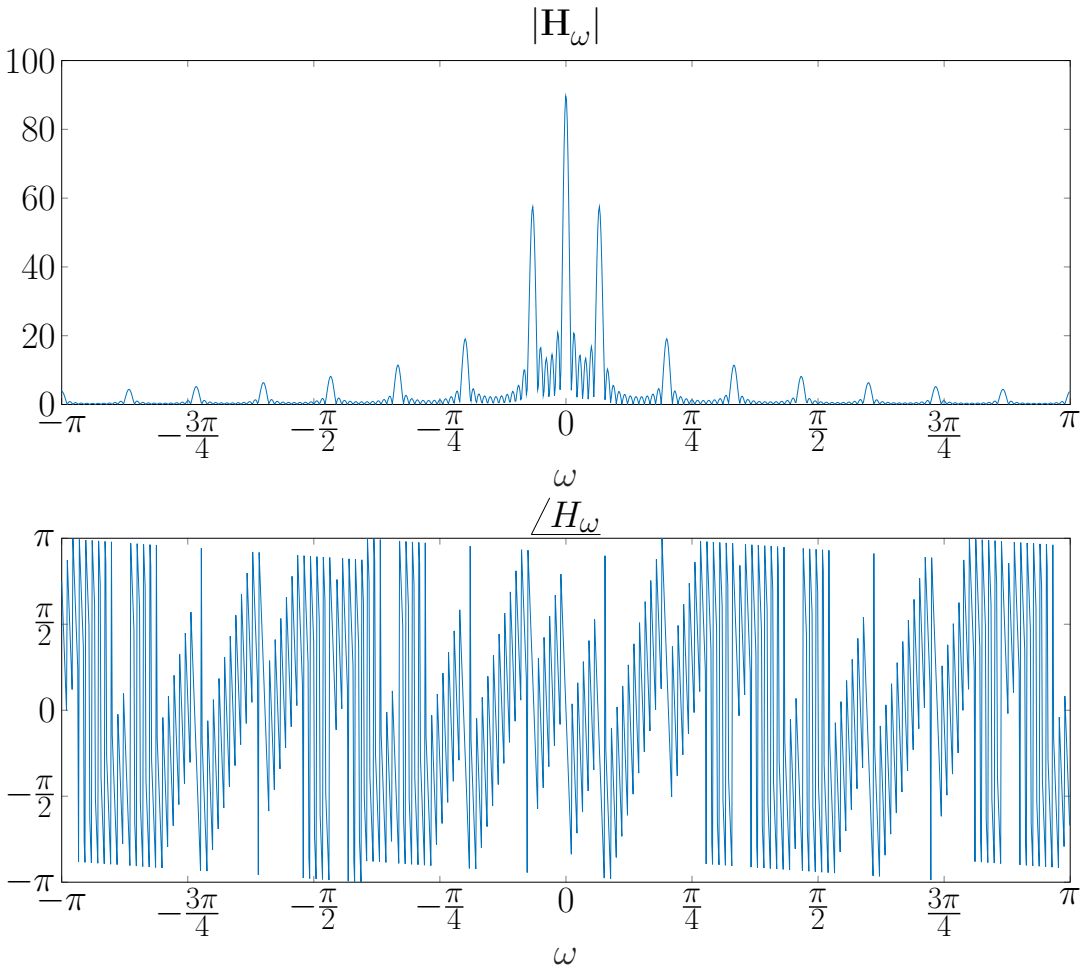


Figure 3.15 – (Top) Magnitude and (Bottom) Phase of a theoretical square wave conformed by a finite sum of square pulses, used as a model for the pad traces left by the logging tool on our Borehole Image Data logs.

Nonetheless, equation (3.25) has only one dimension, while our image has two dimensions. Even though we could overcome this problem by applying (3.25) line by line to our image, this would be impractical and would make the final processing much more time and resource consuming than by just applying a single 2-D filtering process to our image.

As it was stated before in this Section, we can fairly assume that the pad traces are periodic only over the horizontal axes. This means that we can assume that the value of these traces does not depend on the vertical position or, in other words, that they are just a vertical repetition of a horizontal square wave (as shown in Figure 3.13).

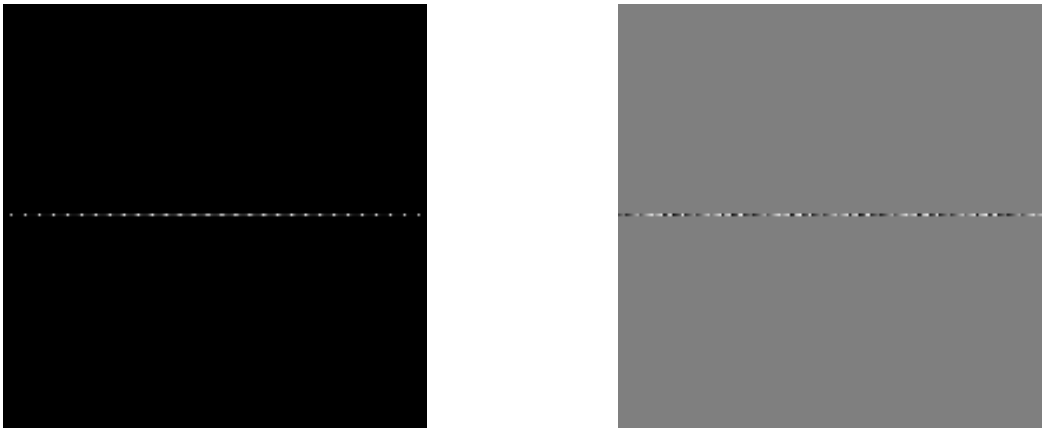
The previous statement allow us to rewrite (3.19) as the separable function

$$h_{xy} = h(x) \cdot 1(y). \tag{3.26}$$

It is also well known that the fourier transform of a constant with value 1 is a dirac delta Bracewell and Bracewell (1986). Using these two principles we can expand (3.26) and obtain the final version of the 2-D fourier transform of the pad traces we are trying to suppress, which can be written as

$$H_{xy} = H_x \cdot \mathcal{F}\left[1(y)\right] = \sqrt{\frac{2}{\pi}} \frac{\sin\left(\omega_x D \frac{N}{N_p}\right)}{\omega_x} \cdot \sum_{k=0}^{N_p-1} e^{-j\omega_x \frac{N}{N_p} \cdot \left(\frac{D}{2}+k\right)} \cdot \delta(\omega_y). \tag{3.27}$$

An example how the 2-D spectrum (magnitude and phase) of (3.27) looks when $D = 0.5$, $N = 180$ and $N_p = 6$, is shown in Figure 3.16.



(a) (b)

Figure 3.16 – (a) Magnitude and (b) Phase of the 2-D fourier transform of the pad traces signal expressed on (3.27).

3.4.2.2 Automatic pad period detection

The final expression of the 2-D fourier transform of the traces left by pads represented on (3.27) depends on three basic parameters:

1. N: The horizontal size of the input image.

2. N_p: Number of pads of the logging tool.

3. D: Relation between the size of the pad and the total period of the square wave model, as shown in (3.14).

While the first parameter, N, is a constant and well-known value (number of columns of the image); both N_p and D are unknown and depend on the tool used during logging procedures. We know, however, than on regions of our image U_{xy} where the pad traces are very noticeable, its 1-D horizontal spectrum magnitude should present a very sharp peak around the value of ω equivalent to the period of the square–wave model of the pads or, in other words, to the space between two consecutive pads.

Thus, if we extract the 1-D fft for each line of our image, and average these values for the whole image, we ought be able to deduce the period T of the pads traces wave. An example, applied to the same interval of borehole image shown in Figure 3.13, is shown in Figure 3.17.

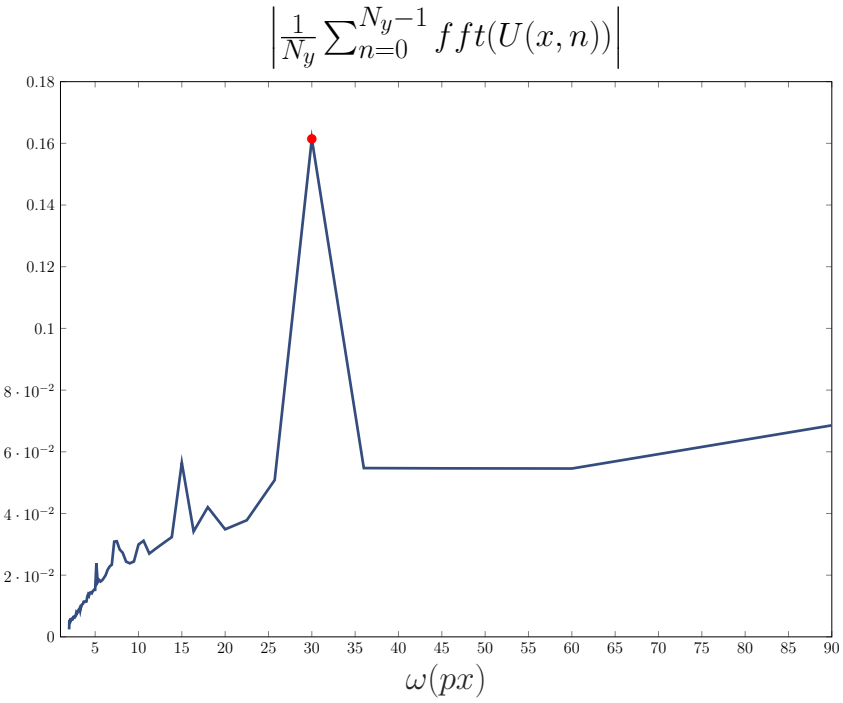


Figure 3.17 – Average of the magnitude of the FFT obtained from the image U_{xy} for each line.

As it can be seen in the previous image, the most prominent peak is located at $\omega = 30px$[5]. This means that, for the analyzed image, each pad trace is separated 30 pixels from the next one. It is now possible to calculate the total number of pads, N_p by using this value and (3.15). In this case, the value of N_p deduced from this analysis is 6, which can be confirmed by visually checking the previous image.

[5]Notice that in this case the horizontal axis has been normalized and discretized so that it represents values of pixels of the input image.

The coefficient D is much more difficult to obtain by image processing. It could be possible to estimate it by using fitting or segmentation methods, by checking different coefficients of D and analyzing the fit of the resultant square-wave model, line per line, however this has not been tested in our case. Instead, we allow the user to modify this coefficient, allowing them to use it as a tunning parameter that controls the quality of the filtering process.

3.4.2.3 Results

The following figures show the results of applying this method to remove the pad traces on an ultrasonic borehole image log provided by PETROBRAS. Only the intervals that displayed pad traces on the original images are shown as the filter does not interfere in the image where no pad traces appear. The left images that appear on Figure 3.18a, Figure 3.18b, Figure 3.18c and Figure 3.18d show different intervals of the original borehole data, while the right images display the data after the application of the filter.

As it can be seen the filter shows notable gains by preserving the most important lithological features of the rock behind the pad traces while suppressing the pads, thus making the task of further texture and lithology characterization of the borehole much easier.

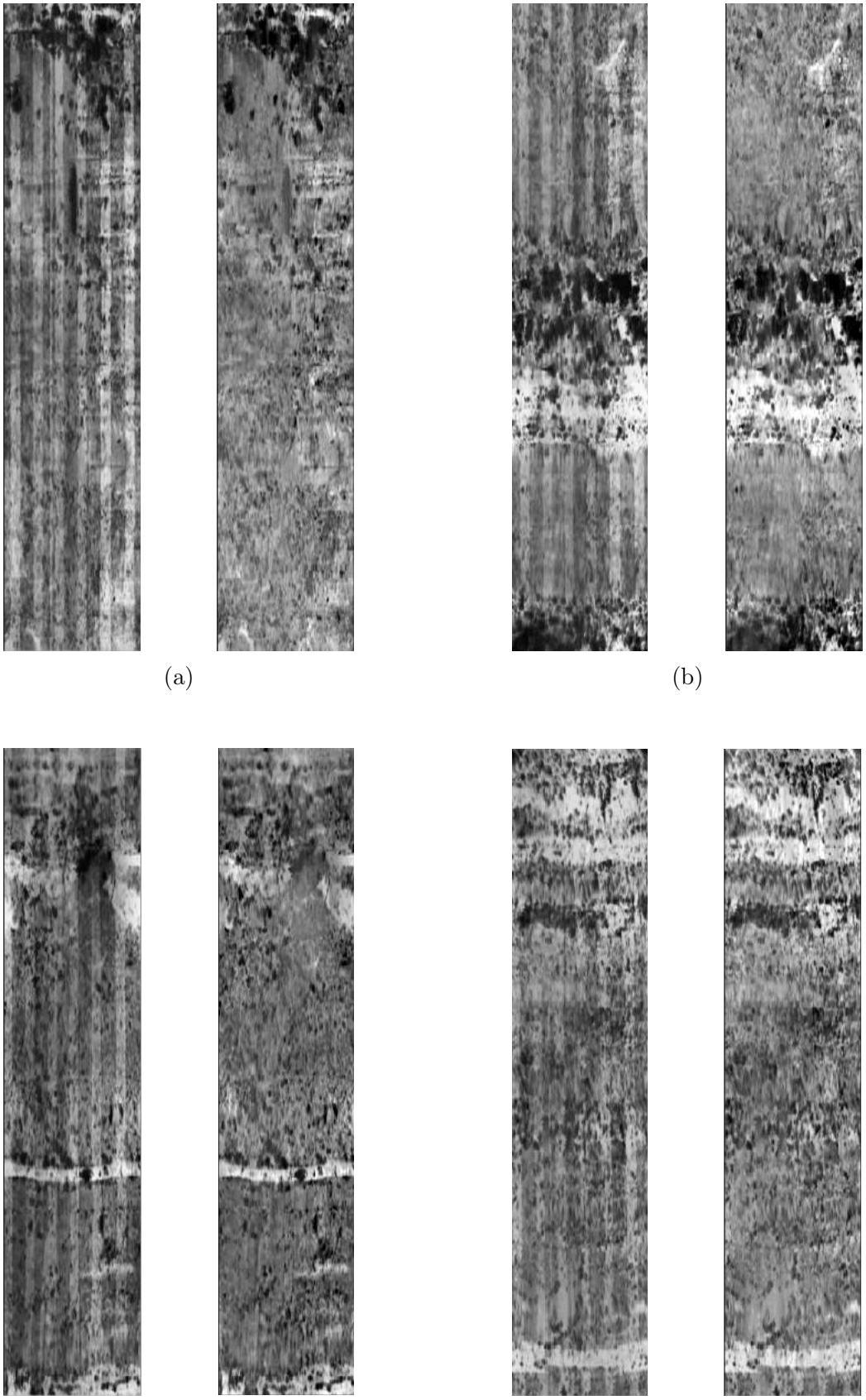


Figure 3.18 – Result of applying the proposed method to different intervals of an Ultrasonic Borehole Image log. For each one of the four intervals (a), (b), (c) and (d) the images shown in the left represents the original data, while the ones on the right represent the result after applying the filter.

Chapter 4

Deep learning and applications

4.1 Brief introduction to artificial neural networks

The idea of an artificial mechanism able to simulate biological neural systems is not new. McCulloch and Pitts (1943) introduced the idea of an artificial neuron, whose aim was to recreate the behaviour that biological neurons had. This unit, usually refered to as McCulloch–Pitts neuron, basically computed a weighted sum of each input by a certain interal constant (called weight), followed by a thresholding operation. Using this unit as the essential building block it was posible to create the first artificial neural networks (henceforth, ANN) which were used to implement different simple and complex logical functions, according to the internal weights and thresholding level of each unit. These ANNs, however, had a major drawback: as the weights of each neuron were fixed, the net could not learn nor adapt to new examples.

During the following years, Hebb (1949) and after him Minsky (1954) proposed different modifications to be made to the learning algorithm, so that the network would become a much more flexible system, thus allowing the net to adapt to new conditions and examples. Rosenblatt (1958) set a milestone in the artificial intelligence (henceforth AI) world by proposing the perceptron model and the perceptron learning law, which proved to converge the feature space for several different pattern classification problems.

Even though, mathematically speaking, the foundations of artificial neural networks had been set up to that point, it was not until the development and improvement of computational systems in the 1980s that the use of ANNs for pattern recognition problems started to grow (Anderson and Rosenfeld, 2000). From that moment on ANN have been widely used in all fields of science that require any type of signal processing or pattern recognition, such as in medicine for cancer detection (Esteva et al., 2017; Bhardwaj and Tiwari, 2015; Lisboa and Taktak, 2006), engineering (Esfe et al., 2015; Ali et al., 2015) or physics (Carleo and Troyer, 2017; Kumar et al., 2015), to name a few.

ANNs are called artificial because they are inspired by the structure of the biological neural structures found in nature, i.e. like the ones found in our brains (Nielsen, 2015). Even though in the past years these models and the different types of structures have evolved and spread enormously, they are still very far from showing the performance that natural systems, mainly because it is almost impossible to simulate the number and structure of the interconnections of neurons present in any real neural system.

Although our current ANNs are very small in size when compared the brain of any animal, they are already displaying some of the features of pattern recognition and artificial intelligence that they aim to emulate from biological systems (Yegnanarayana, 2009). One of the main characteristics that make neural networks so interesting is their flexibility: NNs adapt to new circumstances and learn how to deal to unexpected scenarios, even when no pre-programmed instructions are given. In other words, one of the most valuable things about neural networks is that they are fuzzy systems that can work with noisy and probabilistic data. Being able to create fuzzy systems that are not completely logic was a major step in the artificial intelligence field (as previous AI systems were based on logical computations which made them very sensitive to noise).

Current neural networks are composed by gathering several artificial neurons and training them for different purposes (classification, regression, pattern recognition, etc) (Kröse et al., 1993). There are two major structures that define the type of direction of the data inside the net: feedforward or recurrent. Feed-forward neural networks (FFNN) commonly present a layer-stack type of structure and always have a directed acyclic graph, which means that the information contained on each layer can only be passed forward on the net, but never backwards. On the other side, Recurrent Neural Networks (RNN) might let the inner information to flow in different directions (forward, backwards and even recursively), which allows these nets to present dynamic temporal behaviour and even long/short term memory. Most data processing applications use FFNN as they are usually more predictable and more easily applicable to classification and pattern recognition problems.

Based on these two different types of data-flow inside the ANN, several inner structures have been proposed. A complete list of most known types of ANN is presented in Figure 4.1. Further detail about each type of network and their applications can be seen in the source of the image presented below: Veen (2016). For more information about simple neural networks, their use and applications see Egmont-Petersen et al. (2002); Patterson (1998); Cochocki and Unbehauen (1993); Peretto (1992).

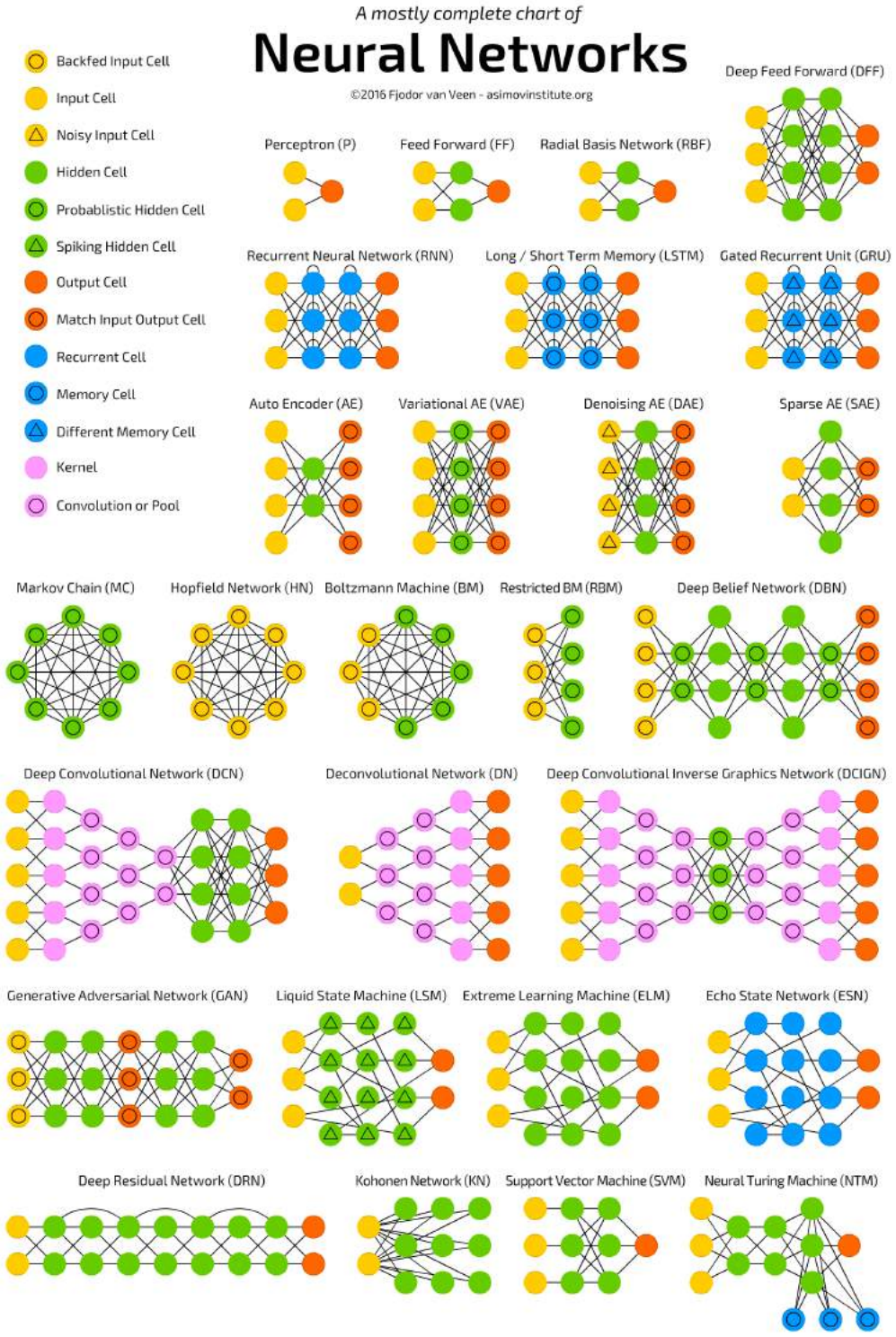


Figure 4.1 – Chart of all different 27 types of ANNs structures found nowadays. It can be found a total of 13 different artificial neurons types, each one implementing a certain specific function. Input cells behave as bridges between the internal layers of an ANN and its inputs, while output cells do a similar job at the output of the ANN. Hidden and convolutional cells extract features from the data inputed to them, which will be later used for complex purposes (such as classification or regression). Memory and recurrent cells allow the net to keep track of previous values present at different points of the net, so that the ANN itself can present some sort of short-term memory. Combination and concatenation of these times of cells allow us to create all the types of ANNs shown in this illustration. Source: Veen (2016).

4.2 Deep learning

One of the most spread ANN structure (Hassoun, 1995) consists of simply stacking three fully connected neuron layers in a backpropagation/feedforward architecture, so that the first layer is an input layer, the second layer is the hidden layer (which contains all the hidden weights and biases that link the input data and the output variable(s) we are trying to estimate or predict) and the last one is the output layer. This structure is commonly called single hidden layer feedforward neural network (henceforth, SLFNN) and has been considered as one of the simplest types of ANN there is, so far.

It is also possible to stack several hidden layers, one after another, so that the neuron of each layer is connected to all neurons of the previous layer, thus creating what is called a multilayer perceptron (henceforth MLP). MLPs belong to the supervised algorithms class and they are commonly used for both linear and non–linear regression and classification problems. MLPs have been widely and successfully used over the last decades for regression and classification problems that are either non-linear or depend on a great amount of variables (Nazzal et al., 2008; Yan et al., 2006; Cigizoglu, 2004; Gardner and Dorling, 1998). An example of the internal architecture of network with a single output is shown in Figure 4.2.

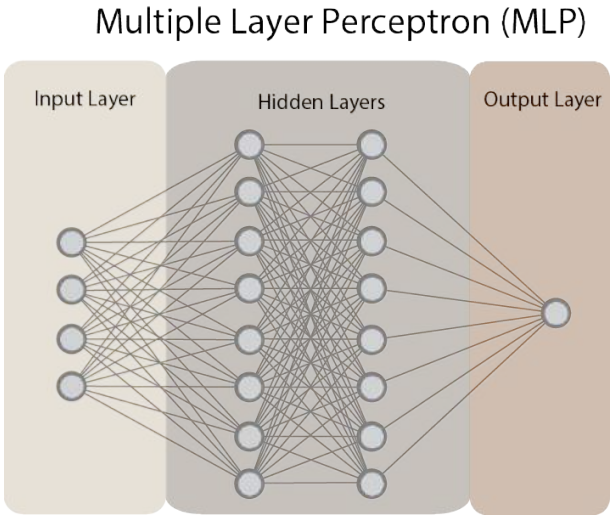


Figure 4.2 – Simplified version of the internal structure diagram of a MLP for data regression –single output–.

Hornik (1991) and Hagan and Menhaj (1994) showed that, from a theoretical and mathematical point of view, even ANNs with a single hidden layer can be considered as universal approximators (Csáji, 2001; Cybenko, 1989). This means that even SLFNN with a finite number of neurons can, at least in theory, be trained to approximate any type of continous function on compact subset $\mathbb{R}^n$, as shown in (Knerr et al., 1992; Huang et al., 2000; Marcialis and Roli, 2005; Auer et al., 2008).

Nevertheless, having an structure able to approximate any continous function does not always guarantees that, for a given particular problem, the training process will lead to

succesfull or acceptable performance results. Indeed, SLFNN usually fail when trying to approximate very complex non–linear functions. When we, as humans, try to understand complex problems we usually attempt to divide them into several simpler problems that can be individually understood and which, when put altogether, allow us to understand how that complex system works.

In machine learning algorithms it is often considered that each layer of the trained net is an approximation of a singular function of the input data. Thus, using multiple layers (singularly or consecutively trained) leads to approximating several different functions that, when combined, might be able to solve the initial complex problem characterized on the input data with greater accuracy than a SLFNN could reach. If one observes any (successfully) trained multilayer ANN it is possible to see how the complexity of the learnt features and functions of each layer is directly proportional to the depth it is located on the net (Eldan and Shamir, 2016). Hence, the first layers of the network usually capture simple and basic features of the input data, while the last layers present complex patterns of the relation between each of the previously connected layers and the input data.

The process by which we humans understand a spoken sentence (Allen, 1994; Lippmann, 1997; Greenberg et al., 2004; Hickok and Poeppel, 2007; Rauschecker and Scott, 2009) is a good analogy of the increase of complexity of deeper layers inside a deep network: first the brain delegates to the auditory system the task of sensing an incoming sound; then the cortical structures of the brain involved in hearing processes have to determine whether than sound is human or belongs to any other kind of natural or artificial sounds; once it has been determined that the sound comes from a human being, the brain must process the changes in tone, frequency and silence gaps in the incoming sound in order to form complex data which we call sentence; lately, it is necessary to give each of these units of sound (words) their right meaning (which also involves detecting the language on which the emitter is speaking, in polyglot brains), thus allowing us to understand the concept that the emitter of the sentence wanted to express. It is obvious that the cortical networks that allow us to process the meaning of each word in the brain are far more complex than those dedicated to simply detect the nature of a sound (or whether it is or not life–threatening).

These architectures formed with several partially or fully connected layers are commonly called deep networks and the field of AI dedicated to its study is called deep learning or hierarchical learning (Bengio et al., 2009; Deng et al., 2014). It is worth noticing that Deep Learning does not necessarily involves a deeper understanding of the input data processed by the net, but rather the techniques and methodologies that automatically learn deep features and functions in order to solve a particular complex problem (i.e., classification, regression, etc).

These complexity in the structures created and used in deep learning is not random at all. In fact, if we analyze the systems involved in processing several complex sensory data,

such as speech or visual generation and recognition, we will find that nature has come up with hierarchical structures similar to those used in modern Deep Learning techniques, in order to transform the input waveform level data into meaningful high–level information such as linguistics or object recognition, for the given example.

Over the past years, deep learning has revolutioned the AI and machine learning field (Goodfellow et al., 2016) by making possible the creation of artificial structures able to process complex signals and data from a wide variety of sources, and extract deep features from them that can be more effectively and accurately linked to other properties of our interest (such as classification labels, or other physical properties we might be willing to estimate from the input data). For instance, deep learning has shown very effective results in face recognition (Ranjan et al., 2017; Lu et al., 2017), speech detection and characterization (Zhang et al., 2017; Meng et al., 2017), as well as in medicine (Miotto et al., 2017; Rawat et al., 2017; Danaee et al., 2017; Goodwin and Harabagiu, 2017) fields, proving that this approach deals with complex functions in a better way than SLFNNs do.

There exist three main types of deep neural networks (henceforth, DNN), which are (Deng et al., 2014):

1. **Unsupervised or generative DNN**: In this case, no targets (classes nor regression values) are given to the net. The DNN must then automatically extract high–order features from the input data and correlate them with the whole dataset in order to characterize it. The information extracted from this characterization can be also used in a generation model, allowing us to create new synthetic data. Generative adversarial networks (GANs), deep autoencoders (like the one used in Section 6), restricted boltzmann machines (RBM) (Marlin et al., 2010; Hinton, 2010) and Deep Belief Networks (DBN) (Hinton, 2009) fall into this category. For further information about unsupervised deep networks see Erhan et al. (2010).

2. **Supervised DNN**: In this case, targets (discrete or continuous values) are provided to the net, so that its main purpose is to use the high–order features extracted from the input data to create a discriminative model whose aim is to recreate these same targets on its output. By this reason these models are usually called discriminative deep networks (DDN). The discriminative model that links the extracted features and the targets is most of the times based on hidden markov models (HMM) or some type of variation (Eddy, 1996). Nowadays the most commonly used type of supervised DNN is the multilayer perceptron, in all its forms. Convolutional networks, for instance, fall into this category.

3. **Hybrid DNN**: Which is sometimes considered as a special case of supervised learning, as its main objective is to use the outcomes of a generative unsupervised DNN (usually called pre–training process) in a discriminative model able to be used for

classification or regression purposes, as any other type of supervised DNN. Discriminative deep belief networks (DDBN), a combination of DBN and discriminative models, are among the most common type of hybrid DNN used nowadays for several classification purposes (Zhou et al., 2010; Mohamed et al., 2011).

Further reading on deep learning methods and on how deep learning can be applied to signal and data processing can be seen in Schmidhuber (2015) and Yu and Deng (2011), respectively.

4.3 Autoencoders

As introduced before, an autoencoder (henceforth AE) is a type of unsupervised ANN whose objective is to reproduce a set of inputs with the least possible amount of distorsion by creating a hyperspace of features with an encoded version of the original data information. In other words, given a certain input to an autoencoder, its objective is to return a reproduction of the input data from a minimalistic representation, as close to the original data as possible. Although autoencoders may seem naive and useless at first glance, the internal mechanism that autoencoders use to achieve this goal is extremely useful for deep learning and pattern recognition purposes (Baldi, 2012; Ng, 2011).

The simplest version of the internal structure of an autoencoder is composed by three main layers: an input layer with the same dimension as the input data, a hidden layer with a certain number of hidden neurons β and an output layer with the same dimension as the input data. During the training process the autoencoder changes the internal weights of each neuron in the hidden layer, as well as the layer biases, in order to obtain an estimate representation of the input data that minimizes a certain cost function.

Depending on the size and structure of the neurons in the hidden layer, autoencoders can accomplish different tasks. If the total number β is smaller than the total number of inputs, the output of the autoencoder will be a compressed representation of the input. This results in a bottleneck structure that is capable of extracting underlying representations of the input data, retaining only those meaningful features of the input data that are useful for characterizing it, while discarding the information that does not contribute to this task. That is the reason why autoencoders are usually used as data compressors (Tan and Eswaran, 2011; Theis et al., 2017) or denoising networks (Feng et al., 2014; Agostinelli et al., 2013).

A simplified version of the internal structure diagram of a typical shallow autoencoder is shown in Figure 4.3a.

It turns out that the learning process of an autoencoder –tuning the internal weights and biases so that the final output result is similar to the input– results in an approximation of the identity function of the input system. This means that the hidden layer of a trained autoencoder contains useful information about the patterns hidden in the

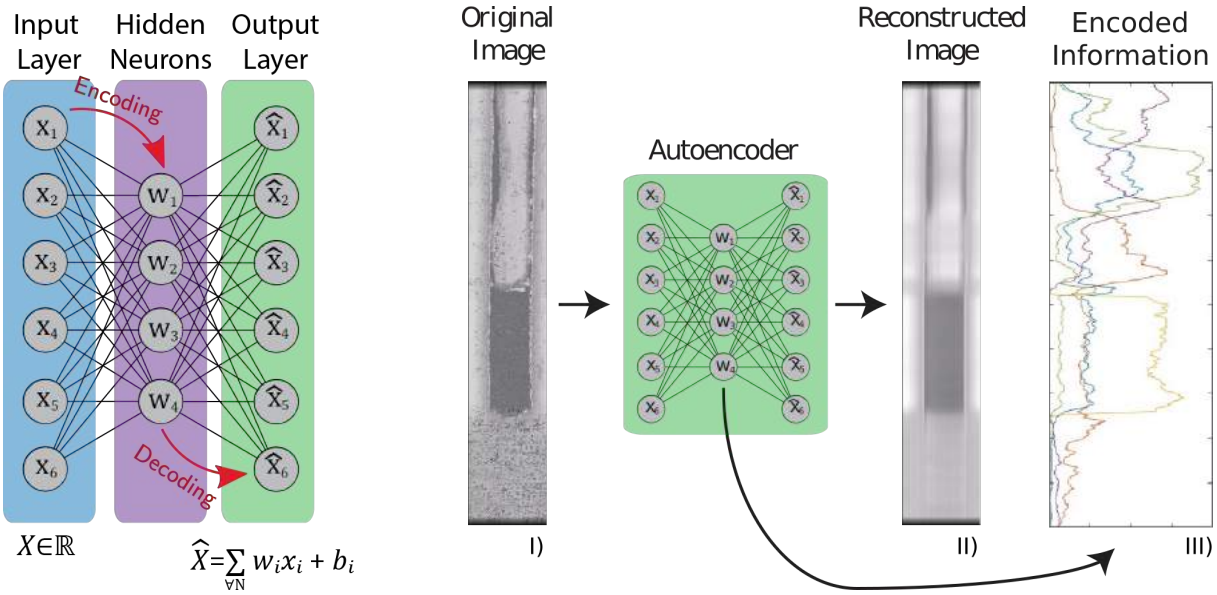


Figure 4.3 – a) Simplified version of the internal diagram of an Autoencoder. The encoding process is performed by the input and hidden layers. The internal values obtained after the hidden layers, for any given input, are the internal features that are used in this work for the petrophysics parameters regression. The decoding process, on the other hand, is performed by the hidden and the output layers. b) Example of autoencoding process for a sample ultrasonic borehole image: (I) original input image; (II) estimated image returned by the autoencoder after the training process. Denoising can be observed in this procedure, although the main objective in our case is to encode the geological information contained in the input image. The input data is encoded by the AEs and can be concatenated to form encoded logs. A sample of the concatenation of the 4 encoded logs for the input sample borehole image is displayed in (III).

input data. By trying to recreate the input data, the autoencoder has somehow learned to produce an estimated version of it, and has done so by estimating a certain group of hidden features from the input data.

The encoded representations obtained from the hidden layers of an AE can be very useful in image characterization. If each image sample introduced to the AE represents a certain object or property (in our case, each sample contains geological information regarding a certain region of the borehole), the AE might be able to characterize the inherent differences between these samples. Then, these encoded versions of the input data can be used for further classification and pattern recognition problems, instead of using the original data. This approach has been applied for medical purposes such as HIV classification (Betechuoh et al., 2006) or breast cancer nuclei detection (Xu et al., 2016).

Nowadays, AEs come and go by different structures and sizes. The structure presented in the previous paragraphs can be repeated and combined forming what is known as an stacked autoencoder. Stacked AEs are a type of Deep Learning technique on which a

series of autoencoders are trained consecutively by using the encoded values extracted from the hidden layer of the previous AE. In this structure, the first AE is trained using the original input data; the second AE is trained using the encoded hidden features of the first autoencoder; and so on. An example of a stacked AE with 4 levels is shown in Figure 4.4.

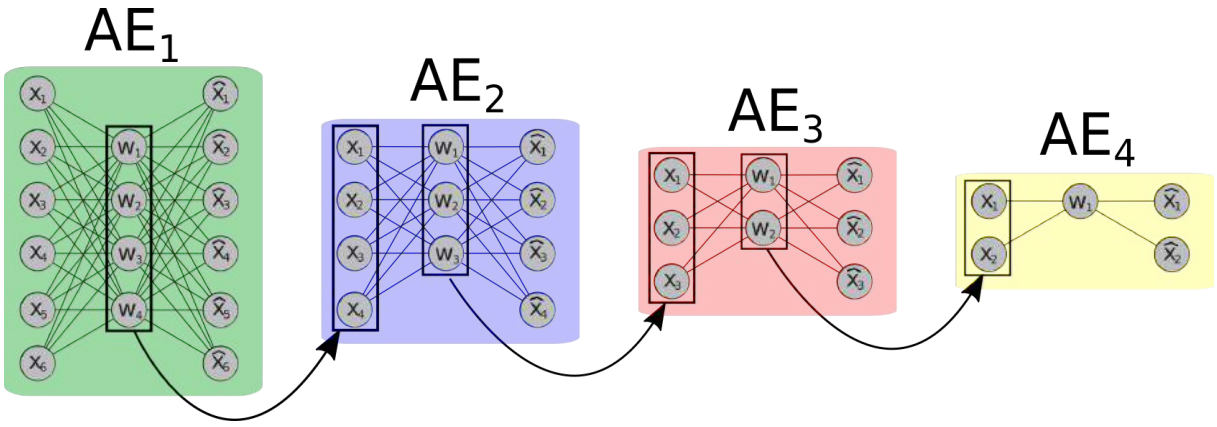


Figure 4.4 – Example of internal architecture of a stack of 4 autoencoders.

Stacked AEs allow us to extract properties of the input data at different levels and scales. Thus, it can be helpful when analyzing images or signals that present different properties at different scales, as it has been shown in recent works (Shin et al., 2013; Xu et al., 2016; Kan et al., 2014). Recently, variation autoencoders have been implemented for subsurface reservoir characterization by Li and Misra (2017).

In this work, we propose to use a 3 level Stacked AE to encode the geological information contained inside borehole images at different scales and use this information in a posterior regression problem in order to estimate permeability and porosity, as will be explained in Section 6. As borehole images are continuous, it was necessary to crop them into slices using a sliding window, so that the information extracted from each slice could be correlated to the petrophysical properties we are trying to estimate. It is worth noticing that these sliced images samples used in our method for each type of borehole image log are represented by $[21 \times 180]$ pixels, which means that the total number of variables to be encoded by the first AE is 3780. The internal structure of each one of these AE was composed, respectively, by: 200[1]., 100 and 50 hidden neurons.

In our method each sliding window produces a single point estimation (porosity or permeability estimated for that specific window), thus allowing us to concatenate the encoded information of consecutive image slices to obtain encoded logs, which are later used to estimate porosity and permeability logs. An example of how this procedure works in a sample borehole ultrasonic image using the first autoencoder is shown in Figure 4.3b,

[1]Using 200 hidden neurons on the input autoencoder (AE_1) net represents compressing the data present in the borehole images using a compression factor of 95%. In other words, this can be seen as forcing the net to learn to represent the borehole images using 95% less amount of information than the original images.

where the original ultrasonic image is displayed in (I), the AE approximation is displayed in (II) and a sample of the concatenation of the 4 encoded logs is displayed in (III). As it can be seen in this Figure, each one of the encoded logs captures a unique property of the input image.

4.4 Convolutional neural networks

Convolutional neural networks (henceforth, CNNs or ConvNets) are a special type of neural network that use convolutional methods to extract features from input data that can be later used for supervised or unsupervised classification, pattern recognition, characterization and regression, among other purposes.

What makes CNNs so interesting is that it takes advantage of the multidimensional structure of the input data, meaning that the high–order features extracted from it depend on the local surroundings (filter size of each CNN convolutional layer) of each sampled value. This is the main reason why CNNs have become so popular over the last decade in different types of deep learning challenges, such as image classification (Peralta et al., 2018; Zhang et al., 2018; Lin et al., 2018; Pradhan et al., 2018; Masood et al., 2018; John et al., 2015; Simonyan and Zisserman, 2014), speech recognition and characterization (Vecchiotti et al., 2018; Choi et al., 2017; Abdel-Hamid et al., 2014; Swietojanski et al., 2014) and even in prognostics medical problems (Li et al., 2018; Rajpurkar et al., 2017) or linguistics problems (Kim, 2014).

Using convolutional layers as feature–extractors instead of fully–connected layers (as in common MLP) has an extra benefit: CNNs are usually more easily trained as they have fewer parameters than its MLP equivalents, because the weights and biases of each convolution layer are connected to a small number of adjacents neurons, rather than to all of them (as it happens in fully connected layers).

CNNs are inspired on biological vision systems, such as the ones found in the eyes and visual cortex of mammals. Hubel and Wiesel (1968) showed that the neurons on cat and monkey visual cortexes respond to small regions of the visual field and that neighboring cells have overlapping receptive fields. Furthermore, they showed that some neurons of the visual cortex of these animals were activated (or fired) only when an image with some specific feature was shown to them: images with vertical, horizontal and diagonal stripes fire, each one of them, different neurons. Their conclusions was that there existed different structures in their visual cortexes which were intrisically speciallized to detect these singular features individually. The main idea behind CNNs is to mimic these biological systems and deep architectures composed by parts that detect individual specific features of the input data.

LeCun et al. (1998) proposed, for the first time, to apply this approach to hand-written digit recognition on 32×32 pixel images by using convolutional methods. In order to train

the weights and the biases of the convolutional layers they used a gradient feed–forward methodology. Their net, called LeNet–5, is used as a common reference until today in the Machine and Deep Learning field. LeNet–5 was composed by 7 different layers: two convolutional layers, two subsampling layers, two fully connected layers and one discriminator layer. Further reading on LeNet–5 architecture and other CNN architectures can be seen in A.

Even though LeCun's paper had laid the foundations of CNNs and how to use them for image classification, CNNs only started to spread in the classification field after 2012, when Krizhevsky et al. (2012) presented their architecture, named AlexNet, in the ILSVR Challenge[2] (ImageNet Large–Scale Visual Recognition Challenge) decreasing the classification error rate down to 15.4% (more than 10% less error than the second classified in the competition).

AlexNet had a simple architecture, when compared to the architectures of current CNNs. It was composed by 8 layers: 5 convolutional layers followed by 3 fully–connected layers.

From that moment on CNNs gained popularity exponentially and can be considered nowadays, without any doubt, the most widely spread methodology used for image classification, characterization and pattern or object recognition problems. As the years have gone by, the nets architecture and complexity has also increased significantly. The winner of the 2016 ILSVR challenge had a total of 152 layers, an increase of 1800% when compared with the 8 layers of the AlexNet presented by Krizhevsky et al. in 2012. Figure 4.5 shows the evolution of the classification error rate in the ILSVR Challenge since 2010 up to 2016. All architectures that have won that challenge from 2012 (AlexNet introduction) up to nowadays are CNNs.

An overview on some common architectures for CNNs published in the last years can be seen in Appendix A.

4.4.1 CNN Types of Layers

As it was introduced before, one of the main highlights of CNNs is the use of convolutional layers to extract high–order features from the input data. However, this does not mean that CNNs are composed by this type of layers only. In reality, since the born of convolutional networks, there have appeared several types of layers, each one thought to perform a specific transformation to data. Designing a D–CNN involves combining these layers in a structure that is able to extract the deep features of the input data, which can be later used either in discriminative or generative model, or simply to characterize the data.

[2]The ILSVR Challenge is a competition where each team is required to do different image processing tasks such as classification, localization, object detection, etc. The system or model that scores the most on all categories is considered the winner of that year's competition.

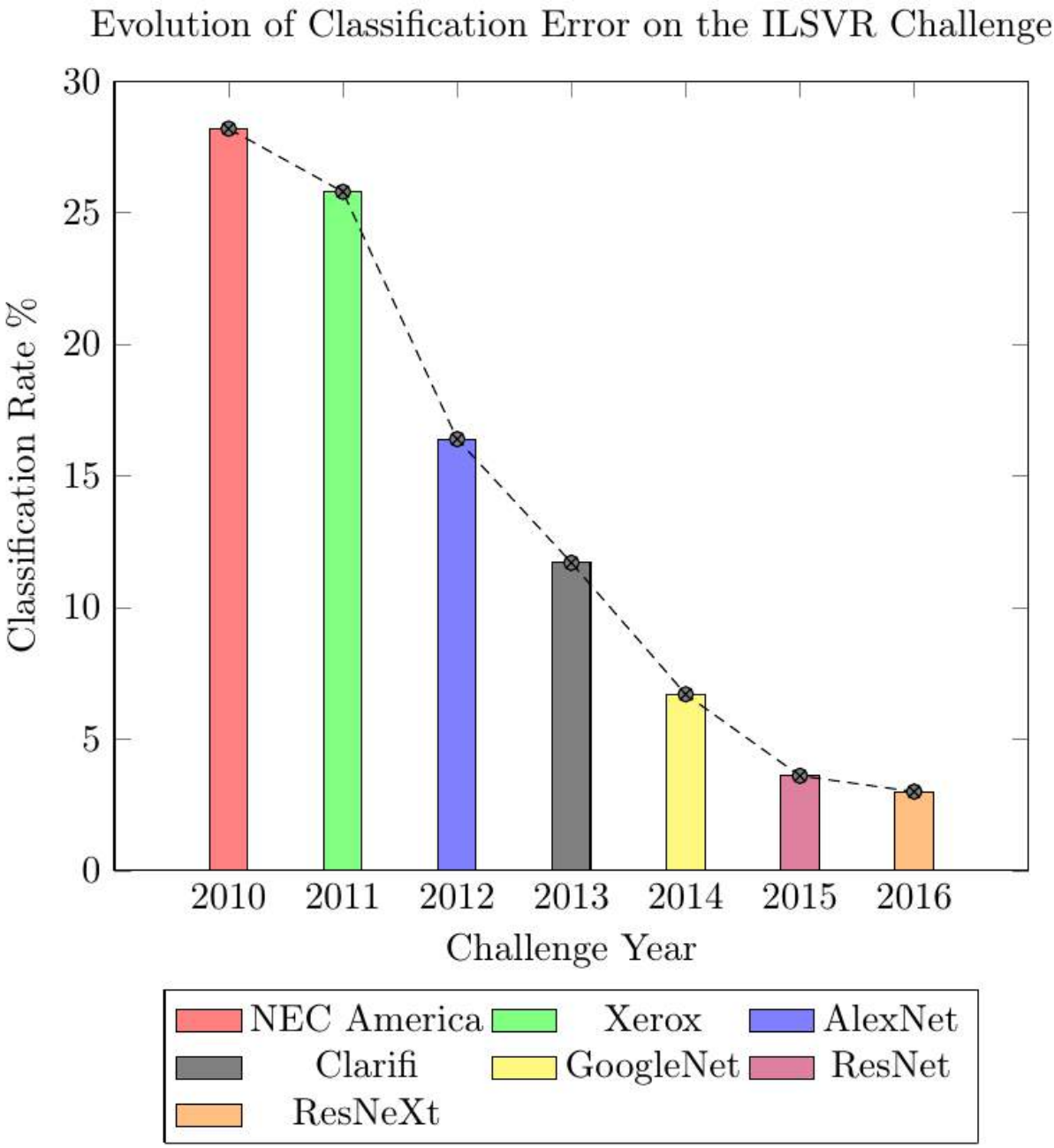


Figure 4.5 – Evolution of the Classification Error rate on the ILSVR Challenge over the years, since 2010 up to 2016. As it can be seen the introduction of AlexNet in 2012, as the first CNN, revolutionized the image classification field. References on some of these architectures can be found: AlexNet (Krizhevsky et al., 2012); GoogleNet (Wu et al., 2014); ResNet (Garson et al., 2002); ResNeXt (Java)

4.4.1.1 Convolutional Layer

Convolutional Layers are the heart of CNNs. The purpose of this kind of layer is to extract volume features from the input data while preserving its multidimensional relationships. While fully–connected layers analyze each value, sample or pixel without taking into account its surroundings, the weight and bias value of each convolutional layer

inside a CNN do depend on their surroundings. This preservation is extremely useful when dealing with multidimensional data, such as images or volumes, on which each pixel or voxel has a strong correlation with the pixels or voxels around it.

In order to do so, ConvLayers apply the convolution operator to the input data. Convolution is defined as a mathematical operation of two functions f and g that results in a third function, which can be obtained as the integral of the pointwise multiplication of one function and the other function being translated. For 2–D continuous functions this operation is defined as

$$f(x,y) * g(x,y) = \int_{\omega_1=-\infty}^{\infty} \int_{\omega_2=-\infty}^{\infty} f(\omega_1,\omega_2) \cdot g(x-\omega_1, y-\omega_2) d\omega_2 d\omega_1, \tag{4.1}$$

while its discrete version is defined as

$$f[i,j] * g[i,j] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[i-n_1, j-n_2]. \tag{4.2}$$

Applying the convolution operator on two functions is equivalent to filtering one of them with the other function. Each Convolutional Layer is composed by a set of different filters, also known as kernels or feature detectors, all with the same size, which once trained produce different outputs when filtering the same input image. The outputs of these kernels, which are called activation or feature maps, correspond to different multidimensional features from the input data, and can be used as inputs on the following layers of the net or for any other characterization purpose (such as inputs to a discriminative classifier external to the CNN).

Figure 4.6 shows an illustration of how spacial convolution works. In this case, the original $[7 \times 7]$ matrix is filtered using a $[3 \times 3]$ kernel filter. It is possible to appreciate how the first (green squares) and last (red squares) values of the final Activation Map are obtained: the kernel filter (blue square) applies a pointwise product of a certain region of the original data and itself; the sum of this product is the final value to be set into its corresponding place in the activation map. The location of the result into the activation map is given by the top–left pixel of the region of the original data used in the convolution process. For instance, in Figure 4.6, the first green square in the original data has its top–left pixel at position $[1,1]$, thus its result after processing the region with the kernel filter (2) is set at position $[1,1]$ in the activation map.

It is important to notice that the resulting activation map is always smaller or equal than the original input data (if no padding is applied, as it will be explained later). This is a drawback when dealing with discrete convolutions and discrete data. This is so because both the original data and the filter kernel have limited size and so the convolution operator cannot be applied to any region outside it. Figure 4.7 illustrates this problem: as the values of the original data at position $[-1,-1]$ (top–left corner of the red square) are not defined outside the limits of the original data itself (here represented as question

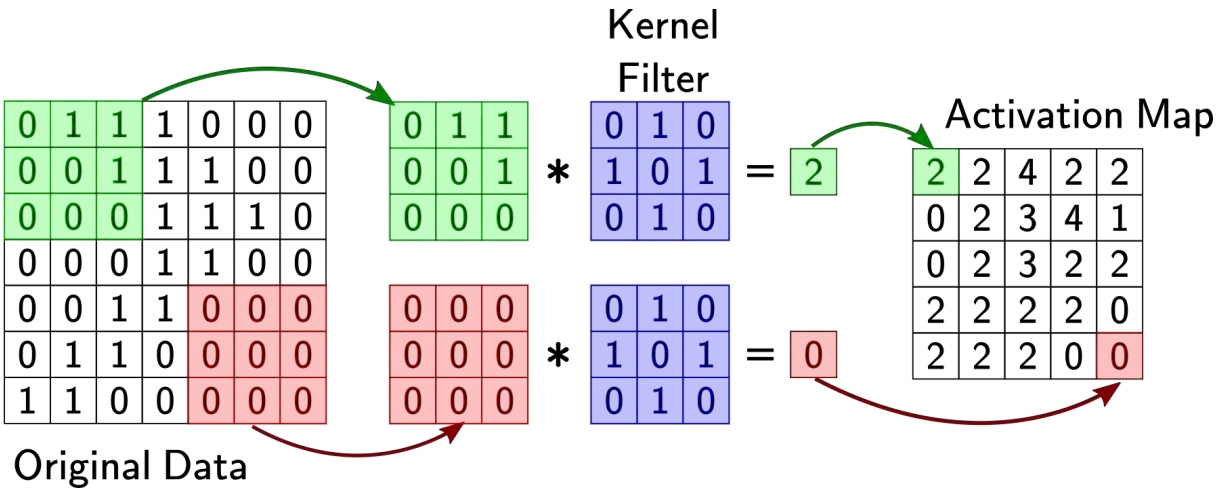


Figure 4.6 – Illustration of how spacial convolution works using a $[3 \times 3]$ kernel filter on a $[7 \times 7]$ image. The colored squares indicate the region being filtered by the kernel filter (blue square) at a certain moment. Notice that the activation map is $[5 \times 5]$ instead of $[7 \times 7]$ due to the padding effect problem explained in this Section.

marks), the result of convolving this region with the kernel filter is also undefined. As it can be seen in the illustration, the final activation map has the same size of the Original Data ($[7 \times 7]$), however its outter pixels are undefined. Thus, the final size of the defined pixels of the activation map is $[5 \times 5]$, just as it was in the case presented in Figure 4.6.

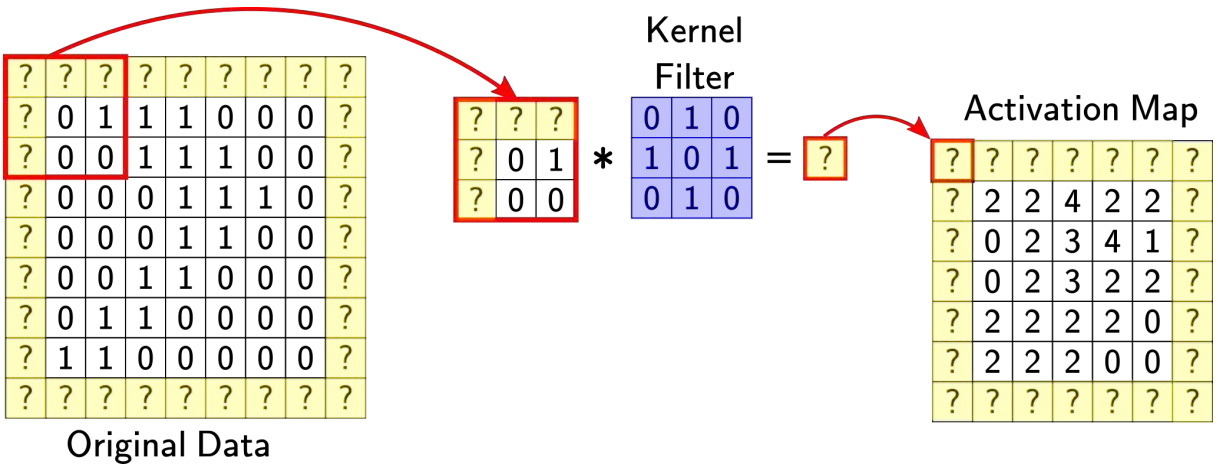


Figure 4.7 – Illustration of why the final activation map after the convolution operation has a smaller size than the original data: the values outside the original data limits are undefined.

Sometimes this reduction of size due to convolution is not convenient. In this case, it is possible to use an operation called padding, which basically adds an outter border to the input matrix (thus increasing its size) with predefined values. These values are not considered to calculate the locations of the values on the final activation map, but rather just to set the undefined values outside the original data limits so the final size of the activation maps can be set to a certain size defined by the user. There exist three

main types of paddings: zero–padding, which sets all outter values to zero; replicate padding, which sets the outter values to the values on the border of the original data; and symmetric padding, which acts as a mirror in the border of the original data. An illustration of the result of applying a 2–level pad using all these three padding methods applied to the same $[7 \times 7]$ matrix used in the previous examples can be seen in Figure 4.8.

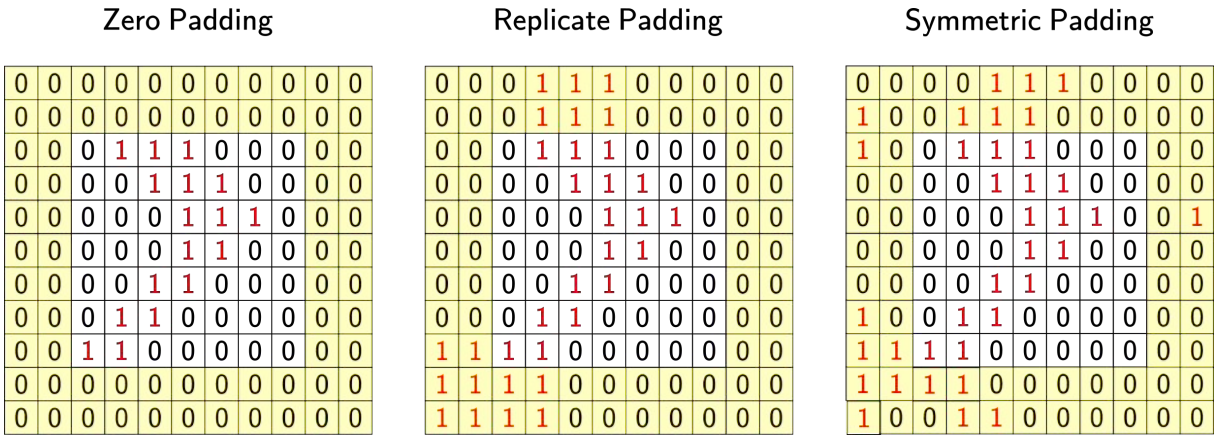


Figure 4.8 – Illustration of the different types of padding methods used to set the final activation map to a certain desired size. In this case a 2–level pad is applied to a $[7 \times 7]$ matrix, using three different methods to deal with its outter pixels. In this image, the yellow squares indicate the location of the pixels outside the original data matrix, which are undefined before the padding is applied.

Apart from the padding, there is another parameter to be considered when dealing with convolutional layers, which also has a major impact on the size of the activation map: the stride. This parameter refers to the number of positions that the filter moves at every iteration. For instance, a stride of 1 means that if at iteration 1 the filter is applied over the region with a top–left corner at $[1, 1]$, the next region to be used at iteration 2 will be $[1, 2]$ (as the filter sweeps the input data horizontally), while if the stride was 2 the next region to be used at iteration 2 would be $[1, 3]$. This process is illustrated at Figure 4.9, where the selected subregions for the three first iterations, using different strides of 1, 2 and 3 clearly produce activation maps with different sizes.

Each ConvLayer is usually defined by the padding $[P_H \times P_W]$ to be applied to the original data, the stride $[S_H \times S_W]$ to be used during the convolution and the size of the layer itself $[H \times W \times C \times D]$, where H and W represent the height and width of the kernels, respectively; C represents the number of channels of the kernels, which must match the number of channels of the input data; and D is the depth of the layer, which represents the number of different kernels that will be contained inside the layer.

Knowing the size of the input data $[M \times N \times C]$ and all these parameters it is possible

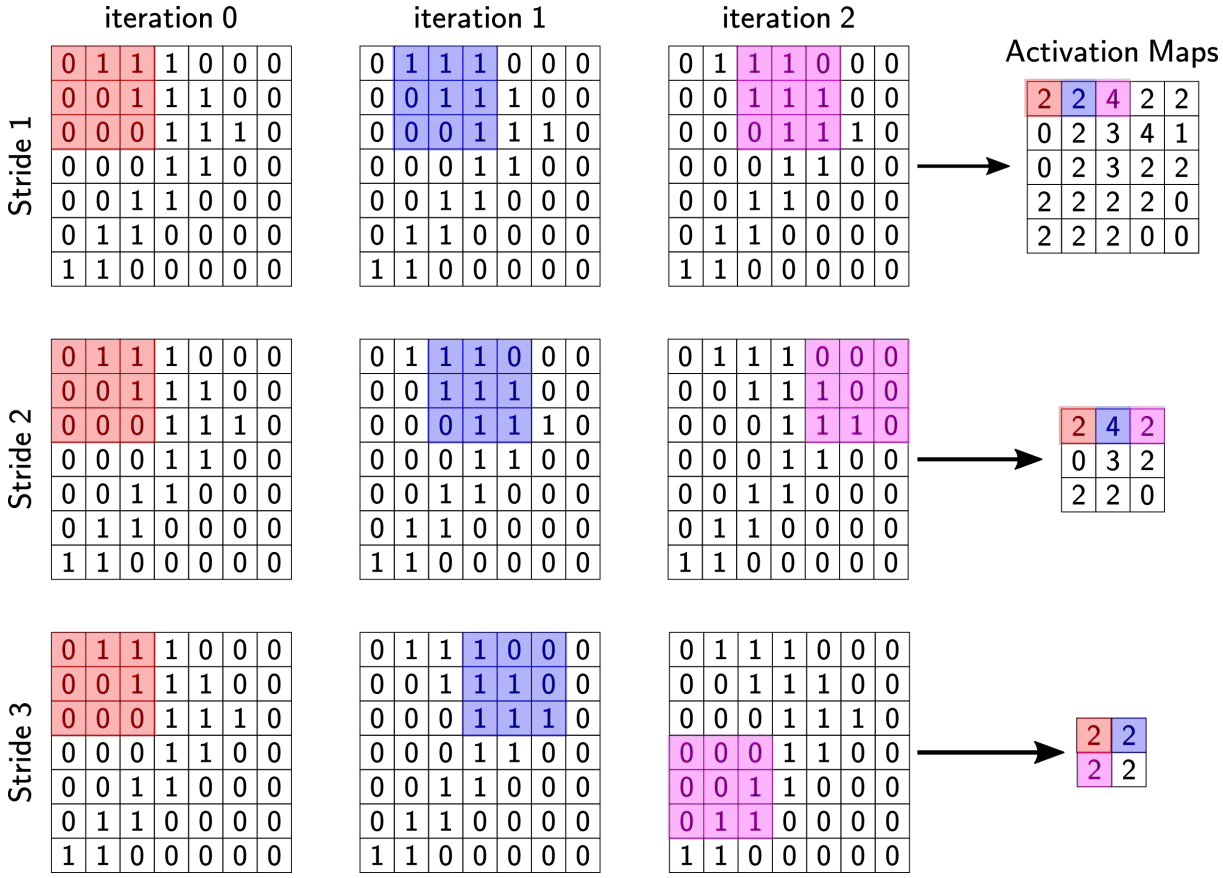


Figure 4.9 – Illustration showing the effect that the stride has on the size of the final activation map. Here the subregions of the original data on which the kernel filter is applied (colored rectangles) are shown for the first three iterations, using three different stride values: 1, 2 and 3. The final activation maps for each value of stride is also displayed on the right.

to determine the size of the output data $[M' \times N' \times C']$ by

$$M' = \frac{(M - H + 2 \cdot P_H)}{S_H} + 1 \tag{4.3a}$$

$$N' = \frac{(N - W + 2 \cdot P_W)}{S_W} + 1 \tag{4.3b}$$

$$C' = D \tag{4.3c}$$

When designing a CNN it is important to determine all the sizes that the activations maps will have along the net, and make sure that their number of channels matches the number of channels of the following ConvLayer that will be applied to them.

Determining the depth of each ConvLayer is not an easy task and usually involves several tests in order to find the right values for them. Nevertheless, most CNN architectures present a simple rule: the deeper the ConvLayer is located on the net, the greater its depth D. This is not a compulsory rule to be followed but rather just an empirical law that has been found to present good results over the years. Still, it is based on the very definition and organization of deep learning and deep networks, whose extracted features

gain complexity the deeper they are located inside them.

The depth of a ConvLayer determines the number of features functions that it can found after the training. As stated before, each kernel filter will perform a single transformation to the input data, thus enhancing a very specific type of feature of it. Different characterization or classification problems, even using the same types of input data, might require different kernel filters, as the features required to characterize the data might be completely different.

Consider, for instance, that we were to find vertical strips on a series of images containing different patterns (strips in different orientations, dots, starts, etc.). It is obvious that the kernel filters on this CNN would have to enhance the features on the input images that are related to vertical stripes, while minimizing all other features (star–related features, dot–related features, etc.). While it was a common practice, before the appearence of CNNs, to manually or semi–automatically design these filters to be used for feature extraction (Randen and Husoy, 1999; Clausi and Jernigan, 2000; Grigorescu et al., 2002; Unser, 1995; Sun et al., 2002), this process demanded either to have great experience on filter design so that the final filter bank was suitable for the required data characterization task, or to use general–purpose universal filters, such as gabor or wavelets, which had the drawback of being less efficient for very specific problems.

Backpropagation and CNNs allow us to obtain these filters automatically, so that after the training process the filter banks of each ConvLayer in the net have been optimized to extract the activation maps that better characterize the features required for the specific problem for which the net was designed.

4.4.1.2 Activation layer

So far we have presented how ConvLayers help extract features from the input data and how these activation maps vary in size depending on the different parameters specified. We also introduced the main concept behind the automatic optimization of the filter kernels inside those layers through the backpropagation mechanism. There is, however, one key concept related to the activation maps obtained by ConvLayers that has not been discussed: the lack of non-linearity.

The effect that a convolutional layer (or any hidden layer on a MLP) with its weights $w_{i,n}$ and bias b_i has on the input data $x_{i,n}$ can be described as the linear operation defined by

$$u_i = \sum_{n=1}^{K} (w_{i,n} \cdot x_{i,n}) + b_i, \tag{4.4}$$

where K is the total number of neurons on that layer (or on every subgroup used on each convolution step) and i is the number of that layer. From (4.4) it is possible to deduce that the outputs of any ConvLayer will be a linear transformation of its inputs. Thus, the operation applied by a net formed with a stack of ConvLayers only would still be a

linear transformation. The problem in this case is that often we are required to work with non–linear functions in the real world, rather than linear ones. It is not possible to fully describe a non–linear function with linear operations only. For that reason it is usually necessary to add a layer to our net which will act as a non–linear operator. As the net gets trained, this operator will also be optimized for obtaining the most suitable features that will help us characterize our problem. This non–linear function is often referred to as the activation function (not to be confused with the Activation Map introduced in Section 4.4.1.1), as its transformation will ensure that only the most relevant part of the Activation Map is passed forward and, thus, "fires" or activates the following layers in the net.

Frequently, each ConvLayer of the net is followed by an activation layer, thus ensuring that each convolutional step (extraction of features) is a non–linear operation. There exist as many activation function as non–linear mathematical functions are there. However, there are three of them which are the most ofently used when implementing CNNs (which can be seen, alongside their equations, in Figure 4.11):

1. **ReLU**: The Rectified Linear Unit function, which acts as a threshold, setting all negative values of the input matrix to zero, and keeping the rest unmodified. It is usually implemented as the maximum between each value and 0: $ReLu(x) = \max_{i \forall N}(x, 0)$. The output values of this function range from $[0, +\infty]$.

2. **Sigmoid**: Sigmoid function, sometimes called logistic function[3], saturates the values that are far away from 0, so that the output values range between $[0, 1]$. Its major drawback is that it might saturate the gradient for values near the limits (away from zero), which means that the training process would be useless for that specific layer. Apart from that, the outputs are not zero–centered, which might cause dynamics problems (fluctuations) in the errors during backpropagation, which might likely lead to unstabilities.

3. **TanH**: Hyperbolic Tangent function. This function acts as a type of saturation function, making the output values range between $[-1, +1]$. In practice, this function has the same properties as sigmoid function, except that its output is actually zero–centered. This is the main reason why *tanh* layers are preferred over sigmoid layers.

An illustration of a convolutional (or a hidden) layer followed by a non–linear operator can be seen in Figure 4.10.

Throughout the years several authors have argued the benefits and drawbacks of different types of activation functions (Glorot and Bengio, 2010; Nair and Hinton, 2010; Maas et al., 2013). On the same paper where Krizhevsky et al. presented the architecture

[3]The Sigmoid function is a special case of the logistic function for $L = 1$, $x_0 = 0$ and $k = 1$.

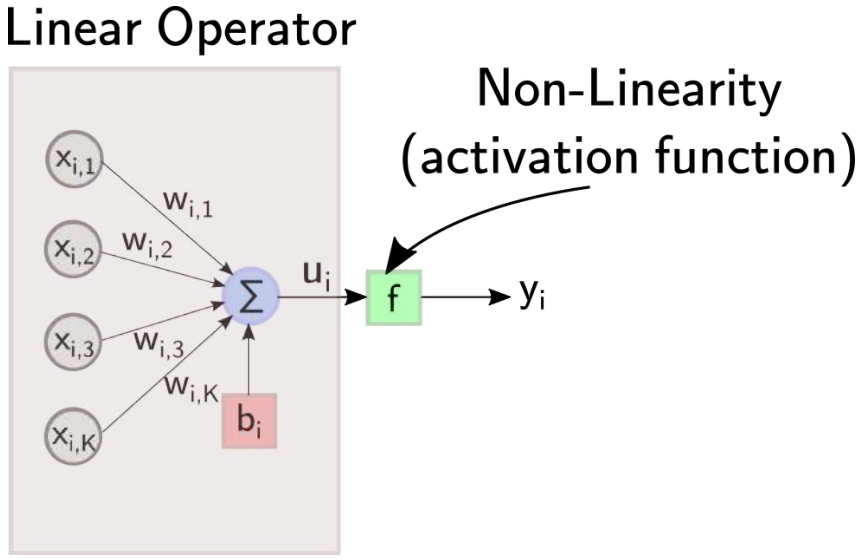


Figure 4.10 – Illustration showing the linear operation carried on inside a ConvLayer on a CNN or a hidden layer on an MLP, where the inputs $x_{i,n}$ are multiplied by the weights of the layer $w_{i,n}$ and added to the bias of the layer b_i; and the following non–linear operation that takes place with the activation function f, resulting in the final output y_i.

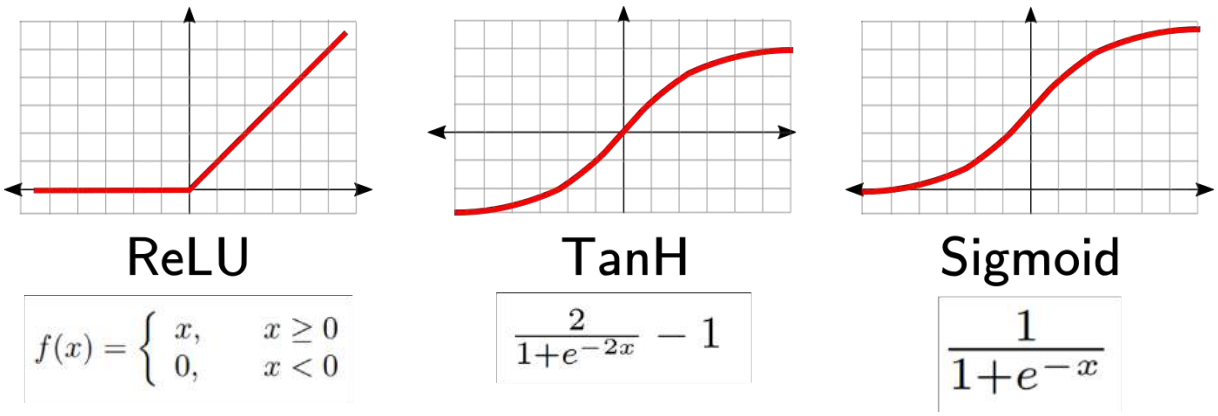


Figure 4.11 – Three main types of activation functions used in CNNs and other types of NNs: ReLU (Rectified Linear Unit function), TanH (hyperbolic tangent) and Sigmoid (a.k.a., logistic function).

and results of the AlexNet (Krizhevsky et al., 2012) (see Section 4.4), they also found that using ReLU layers instead of tanh improved in $\sim$ 6 times the convergence of the net. Apart from this improvement, using ReLU layers does not require computationally expensive processes and its implementation is extremely easy. These are the main reasons why ReLU functions have become one of the most popular activation functions used in Neural Networks. Further reading about these activation functions, as well as their performance on CNNs can be seen in Xu et al. (2015).

Activation layers (and sometimes, specifically called ReLU layers, in case the activation function is assumed to be the ReLU function) do not change the size of the input data, but rather just its values. As introduced before, the most usual architecture consists on

placing an activation layer right after each convolutional layer on the net.

4.4.1.3 Pool Layer

In Section 4.4.1.1 we stated that, even though there does not exist a single universal rule which explicits how the depth of each ConvLayer must change according to its position on the net, it is a common practice to increase this value the deeper the ConvLayer is located inside the net, so that more complex features can be extracted from the data. However, increasing the depth of the ConvLayers steadily inside the CNN comes at a price: if the size of the filter kernels is kept constant, the number of parameters to be optimized (weights and biases) grow exponentially with the number of ConvLayers of the CNN.

Another issue to be discussed about the activation maps created by Convolutional + ReLU Layers is spatial invariance. Since the dawn of object recognition and characterization field there has been a major effort on trying to find the right group of functions or methods to be able to extract features from input data that could universally be used as descriptors of its properties, thus making the characterization task much easier. Among all different properties that these functions should held there is one in particular that attracted the attention of several authors in the last decades: rotation and location invariability.

To us, humans, it is pretty clear that two images with the same pattern but rotated different orientations still represent the same type of texture or information, however this task has proved to be a very challenging one to be taught to machines. This is the main reason why so many different authors proposed over the last decades many techniques to try this problem (Lowe, 1999; Lazebnik et al., 2006; Brown and Lowe, 2002; Ojala et al., 2002; Tuytelaars et al., 2008).

This is a problem we still must deal with as CNNs work with multidimensional spatially–correlated data. An attempt to prevent rotation and location variability problems consists on training the net with repeated samples that have been shifted, rotated and scaled, so that the network identifies that these features (local position, scale or orientation) are not correlated with the discriminant feature we are trying to reproduce (data class, for instance). However, this obviously increases greatly the number of samples to be used to trained the net, causing the training process to be more time and resource consuming, as well as less efficient.

Instead of using the sample repetition mechanism, and also to kept the number of parameters of the net to grow exponentially with the number of layers, it is common to place what is known as a pooling or simply a pool Layer at different locations of the net, usually after a block (or after some sets of blocks) with a ConvLayer + a ReLU Layer.

Pool layers were mainly conceived to reduce the dimensionality of the activation maps inside the net, in an attempt to avoid an excessive amount of parameters, as well to make

the training process more efficient. However, it was soon found that this reduction of dimensionality also introduces an extra non–linearity to the net, allowing it to capture features from different scales. These layers work on subregions of the input data, in a similar way ConvLayers do, applying a certain function to it. Just like activation layers can use different activation functions, Pool Layers can vary from CNN to CNN depending on the method they use to fulfil their purpose. The three main types of pooling functions on which pool layers can be based are: max–pooling, subsampling and sum pooling.

Max–pooling operation keeps the maximum value of the selected subgroup of the input data only, the subsampling method passes on the average value of that subregion and the sum–pooling returns the sum of that same subregion. Figure 4.12 shows an illustration of how all these three pooling methods work using a sample $[5 \times 5]$ matrix and a stride of 2.

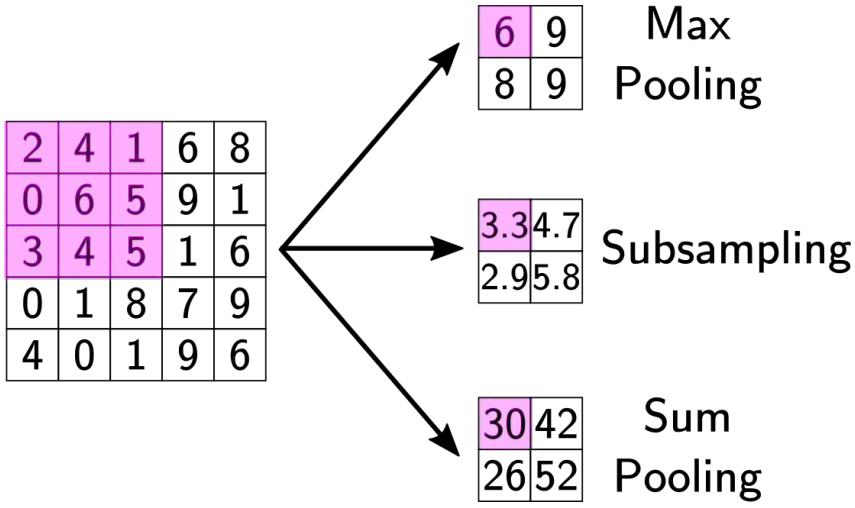


Figure 4.12 – Illustration showing the output of a pooling operation using three different pooling functions: max–pooling, subsampling and sum–pooling. The initial matrix (on the left) is subdivided in groups of 3×3 cells (the first one of these subgroups is highlighted), which are later used to obtain the final result by applying each one of the pooling functions. The division of subgroups depends on the size of the stride. In this case a stride with a value of 2 was used. The influence of the stride value on the size and shape of the final result matrix was illustrated in Figure 4.6. For instance, the first 3×3 group, highlighted, has a maximum value of 6, an average value of 3.3 and the sum of its elements is 30. This is the reason why the highlights cells of the outputs matrices (on the right) are these values, respectively.

Scherer et al. (2010) showed in their work that max–pooling layers performed much better than the rest of them, when tested on the same architecture and for different datasets and types of data. This is one of the main reasons why most pooling layers found in image processing and CNN literature use max–pooling layers instead of subsampling layers.

Just like ConvLayers, pool layers modify the size of the output data, depending on

the size of the window (subgroup) used $[L_H \times L_W]$ and the stride (number of positions to be skipped at each iteration) $[S_H \times S_W]$. As the pooling operation is applied for each slice of data, the depth of the input data remains unmodified. The output size of some input data with size $[M \times N \times C]$ can be calculated by

$$M' = \frac{(M - L_H)}{S_H} + 1 \tag{4.5a}$$

$$N' = \frac{(N - L_W)}{S_W} + 1 \tag{4.5b}$$

$$C' = C \tag{4.5c}$$

Even though Pool layers may have the theoretical benefits introduced before, some authors (Springenberg et al., 2014) have proposed to get rid of them as they consider that their effects are counter-productive in some CNNs classification problems, stating that the lack of pool layers seem to have positive effects on some generative models such as variational autoencoders (VAEs) and generative adversarial networks (GANs).

In the architecture proposed in Section 5.3 for facies classification, we do not use any pooling layers as we found, after a series of trials, that their use was not benefitial for the particular classification problem discussed in this work.

4.4.1.4 Normalization layer

As we introduced in Section 4.4.1.2, ReLU layers are the most widely used activation layers there are. Even with all their benefits, ReLU layers present the main problem of generating unbounded data. This means that the outputs of a ReLU layer (or any other type of unbounded activation function based layer) have a theoretical infinite range of values. The main issue behind this is that any slight error in the weights of these layers may unleash a correlated error that spreads along all following layers of the net (Ba et al., 2016), thus making the net to require much more time to converge. A solution for this problem is to use normalization layers, which limit these activations between certain boundaries. Normalization layers are usually located after linear layers (ConvLayers or fully–connected layers) and right before non–linear layers (such as activation layers).

There exist different types of normalization layers, but all of them have something in common: they change the values introduced to them in a way such that they range inside a certain desired bound (usually from $[0, 1]$ or $[-1, 1]$, depending on the type of layer that follows). Among these different types of normalization layers, the most commonly used are:

1. **Batch normalization layer (B–norm)**: First introduced by Ioffe and Szegedy (2015), this layer reduces internal covariate shift in the input activations by setting their average value to zero and their variance to one. In order to do so, B–norm layers calculate the mean and variance for a randomly chosen mini–batch over each

input channel. It has been found over the years that this simple normalization (1) improves the gradient flow, thus decreasing the convergence time; (2) highly reduces the dependency that CNNs have on initialization values; (3) allows higher learning rates due to the extra stability they create. Further reading on these types of layers and their benefits can be found in Salimans and Kingma (2016); Cooijmans et al. (2016).

2. **Local response normalization layer (LRN)**: Along with AlexNet architecture, Krizhevsky et al. (2012) also introduced on their paper the concept of local reponse normalization applied to CNNs. There they raised the discussion on whether it could be benefitial to implement the lateral inhibition that takes place in our brains, by which an excited neuron has the ability to reduce the activity of its neighbors, creating a contrast in stimulation and thus increasing the sensory perception. In plain words, lateral inhibition is what makes edges with high contrast within images to stand out in our brains. It increases the sensory perception because it produces a certain kind of local normalization on the viewed image, which increases its local contrast and therefore the amount of information our brains can obtain from it. Its implementation in CNN architectures takes a small region within the input data and across close kernels, and normalizes it, thus promoting competition between same regions of different kernels and by doing so encouraging the net to keep only high peaks of activations rather than continuous regions with constant activation values. Krizhevsky et al. (2012) themselves present in their paper the increase in performance that including these normalization layers had on their classification problem.

In the CNN architecture for lithological facies classification proposed in this work we use batch normalization layers between each convolutional layer and the ReLu layer after it.

4.4.1.5 Fully–connected layer

A fully–connected layer (henceforth, FC) is a type of layer similar to a ConvLayer, with the main difference that all its weights are directly connected to all input activations. An FC layer is a common type of hidden layer, like any layer in an MLP nertwork structure. FC layers are usually found at the end of the structure of a CNN, as they work as the final step that links the high–order features extracted through the net with the values we are asking the net to predict for us.

FC layers in a CNN act as a bridge between these features and the targets by learning non–linear combinations of them in order to produce the most accurate prediction. The last FC layer of a CNN must always match the number of target classes we are aiming to reproduce. For instance, if we are trying to classify images from 6 different animals,

the final FC of our CNN must have 6 neurons; while if we were to predict the orientation that hand–written digits have on certain images, the last FC of our CNN should have 1 single neuron.

Often a group of FC layers are stacked together at the end of the CNN, in order to gently reduce the number of features up to the number of hidden neurons required at the last FC layer, although this is not a compulsory method.

4.4.1.6 Dropout layer

This type of layer was thought as a way to reduce overfitting on ANNs. Overfitting refers to the issue caused when a succesfully trained net does not perform well when dealing with new data, as it was biased to solve the specific problem introduced by the sample data used during the training stage.

The term dropout refers to the procedure of randomly dropping out or cancelling a certain amount of weights from the neural network that would otherwise be taken into account when computing the final output. This methodology was firstly introduced by Srivastava et al. (2014) and it has been used over the last years as a simple yet effective way to reduce the effects that overfitting have on NN.

The main aim of dropout layers is to simply set to zero the values of certain proportion p of randomly chosen positions of the input activations. This simple procedure forces the net to be redundant, which means that even when some activations are dropped out the classification results is still right. By doing so the net is trained in a way that it makes sure that it does not get too fitted to the training data, and it is versatile enough. Figure 4.13 shows an illustration regarding how dropout layers work.

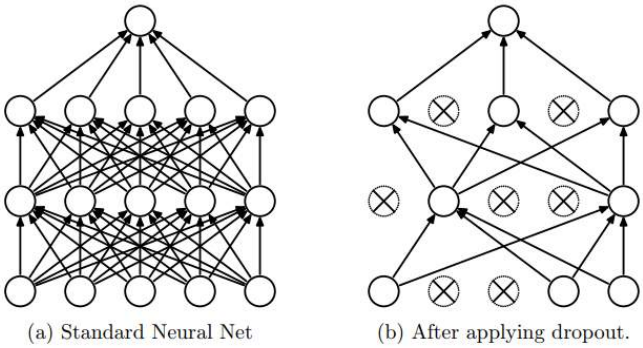


Figure 4.13 – Illustration showing how dropout layers work: some of the values of the input activations are set to zero, thus forcing the net to be versatile enough to produce good results even without them. In this illustration, a comparison between (a) a standard FC NN, whose weights are all connected from layer to layer and (b) a layer where some weights have been dropped out (cells with a cross inside). These dropped cells do not contribute to the final result of the layer. Source: Srivastava et al. (2014).

Srivastava et al. (2014) proposed in their original paper to use a dropout layer after each fully–connected layer. More recently, Park and Kwak (2016) studied the effect these layers have when used after ConvLayers.

4.4.1.7 Loss layer

The most common final layer of a CNN is a so–called loss layer. The main purpose of this layer is to determine the penalization of each deviation between the prediction of the net and the target value of each sample. In plain words, loss layers act as a function to the input training targets and the net predictions, returning a value which represents the total weighted error of that training iteration. This error value is then used to evaluate the performance of the net during the training procedure.

Depending on the problem to be solved different loss functions might be considered more suitable:

1. **Softmax loss layer**: This type of layer uses the softmax function, which is a generalization of the logistic function. This function basically returns the probability distribution of a certain vector over K different mutually exclusive levels or classes. Softmax layer is commonly found on classification problems where the activations of the last FC layer must be converted to N–class probabilities in order to obtain the final prediction (which is set to the class with the higher value). The sum of all the output values of this layer is always one.

2. **Sigmoid cross–entropy loss layer**: This type of layer applies the concept of cross–entropy and information theory (Rubinstein, 1999) to calculate the error of each prediction, for a multiclass problem, assuming that all levels or classes are independent. This function penalizes probabilities of correct classes only, rather than taking into account all classes probabilities, which makes the final error value independent of how the remaining probability is split between the rest of the classes.

3. **Euclidean loss layer**: This type of layer uses the sum of the euclidean distance between the predicted values and the targets as the layer error. It is most suitable for regression problems where it is necessary to fit unbounded real data that can range from $[-\infty, \infty]$.

In the CNN architecture for lithological facies classification proposed in this work (Section 5) we use a softmax loss layer right after the last FC layer, as the classes in our problem (litological facies to be predicted) are mutually exclusive.

Chapter 5

Reservoir lithofacies characterization using deep convolutional neural networks and borehole image data logs

5.1 Reservoir's facies characterization

In Section 2 the concept of reservoir characterization, as well as its importance, was introduced as one of the key concepts to be considered when evaluating a reservoir. Some of the properties that are used to find the unique patterns of each type of rock, such as permeability, porosity or electrical and acustical properties, were also introduced. These properties are oftently used to distringuish between groups of rocks with similar properties, although they cannot be considered enough to define unique universal classes of rock.

A rock body with specific singular appearance, composition, condition of formation, with the same chemical, physical and/or biological features, which makes it distinguishable from any other adjacent rock body is called a Facies (Parker, 1984). The most common types of facies by which rocks can be classified are:

1. **Lithofacies**: Defined by the lithology of the rock, thus its macroscopic physical characteristics, which include its mineral composition, its grain size, its color and its apparent texture. All these physical properties depend on the depositional environment that produced the rocks and so they are usually be used for dating and region tectonic and deposition regime analysis purposes.

2. **Sedimentary Facies**: Defined by the type of sediment that forms the rock. Sedimentary properties do not capture the level of detail contained in lithofacies, however

their value is to be able to supply information regarding the depositional processes that took place and by which the rock itself was created.

Different types of facies are used for differents purposes in order to define the properties of a certain region. In oil industry, on which this work is focused, seismic data is commonly used to find the most likely regions where the reservoirs could be located (if any). After that, these regions are drilled and well–logging tools are introduced to extract different data from them: ultrasonic and microresistivity borehole images, as well as boreline data are used to determine both rock facies and some other petrophysical properties of the reservoir. Borehole image logs also allow petrophysicists to locate fossils and other types of biological structures, which can be used to track environmental and/or evolutionary changes as an attempt to increase the information regarding the geological processes that took place to form the reservoir. On the otherh hand, core plug samples are used to detect the microscopical properties of rocks extracted from the reservoir.

Previous work on automatic facies classification required the use of core, petrophysical logs and borehole images altogether to obtain a reliable prediction (Basu et al., 2002); or provided lithological subjective classifications based on the texture of the data (Chai et al., 2009; Linek et al., 2007; Newberry et al., 2004; Hall et al., 1996). In this Section we present a novel methodology which applies deep learning (using CNNs) to borehole image logs to solve an objective lithofacies classification problem.

5.2 About the data and the training process

The data used in this test belongs to a well called *1–IMET–8–MG* by the Brazilian National Petroleum Agency (ANP), which is located at the São Francisco basin, at 17°25′23.0″ S, 44°38′27.5″ W, near Pirapora City, in Minas Gerais state, Brazil. Its location within the south american continent and Brazil, as well as nearby Pirapora city, can be seen in Figure 5.1.

In this work we have used the ultrasonic borehole image log (UBI), the formation microresistivity imager (FMI) and the lithofacies classification carried out by geologists Claudiano Miranda and Vicente Pimentel. First, all required data was exported from the DLIS file to plain comma–separated text files with a vertical spatial resolution of 0.00508 m. The UBI data was acquired within the range $[465, 1472]$ m, the FMI data within $[490, 1482]$ m and the rock classification was carried out within $[27, 1780]$ m.

Even though geologists are usually able to intuitively classify sections where not all data is available, our model requires both UBI and FMI samples, as well as the ground truth facies classification sample to work. This means that we had to crop our data within the interval $[490, 1472]$ m, thus keeping $193,307$ samples.

The lithofacies data contained 6 different unique lass Labels, on which our method was based. The name of each type of class (in portuguese) and its equivalent in english

Figure 5.1 – Location of the well from which all data used in the rock facies classification section was extracted. Source: Google Maps.

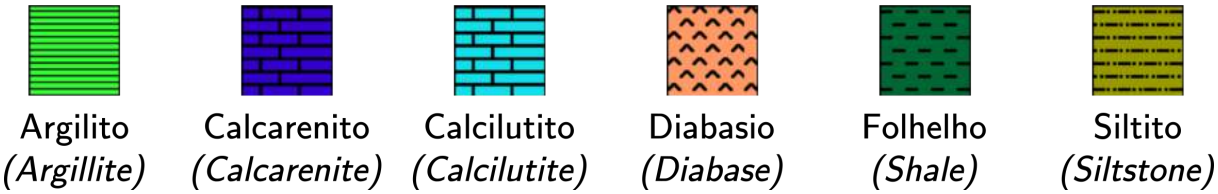


Figure 5.2 – Legend of colours and patterns of each one of the lithofacies classes used in this work.

are shown below, and the legend of colours and patterns of each class used for displaying are shown in Figure 5.2:

1. *ARGILITO*: Argillite
2. *CALCARENITO*: Calcarenite
3. *CALCILUTITO*: Calcilutite
4. *DIABASIO*: Diabase
5. *FOLHELHO*: Shale
6. *SILTITO*: Siltstone

5.3 Proposed methodology for lithofacies prediction

5.3.1 CNN architecture

Our method relies on two different CNNs: one for the ultrasonic image data and another one for the microresistivity image data. The architecture of each one of these CNNs (which will be further explained in 5.3.3) is identical: they contain 5 convolutional layers and a final fully–connected layer, followed by a softmax layer. The first 4 convolutional layers have all of them the same filter size: $[5 \times 42 \times 1]$, but consecutively increasing depths $8, 16, 32$ and 64, respectively. It is noticeable that the sizes of the filters are much greater than any other filter size in the common architectures presented in Section A. Even though the final filter size results from a manual trial and error process in order to find the most suitable values that provided the smallest classification error, it is reasonable that these filters are not squared, as the input images are much wider than higher (the cropped UBI images, and thus the UBI CNN input layer have a size of $[21 \times 180 \times 1]$, while the cropped FMI images, and thus the FMI CNN input layer have a size of $[21 \times 192 \times 1]$).

Due to this difference on height and width of the input images it is essential to insert an extra ConvLayer, right before the first FC layer, that vectorizes the weights. This is the main purpose of Conv5 Layer (which can be seen in Figure 5.3), which takes a $[5 \times 16 \times 64]$ input matrix and applies 128 filters with a size of $[5 \times 16 \times 64]$ so its output is vectorized to a final size of $[1 \times 1 \times 128]$. Thus, Conv5 layer could be also considered as a FC layer as it takes the spatially correlated activations of the previous layer and outputs vectorized activations.

Each one of the first 4 ConvLayers is followed by a batch normalization layer, as an attempt to reduce the dependency on initialization values and the convergence time. In previous tests we found that the error gradient grew exponentially and rapidly, causing the net to fail in the training process, when no normalization layers were used after these ConvLayers.

ReLU layers were introduced right after each one of the normalization layer of these first 4 ConvLayers, as well as after Conv5 layer and FC1 in order to ensure non–linearity to the activations. The output of the last FC layer is a vector with 6 different values, each one corresponding to a certain rock Facies class (see Figure 5.2), which are converted into class probabilities using a softmax loss layer, as Facies classes are mutually exclusive.

5.3.2 kNN classifier

As it is expected that the data of each individual borehole image is not enough to predict the lithofacies of the well, we supposed that the individual activations and class probabilities obtained at the output of each CNN will contain error and misclassification. Thus, once both CNN have been trained, the activations for the training data before

and after the softmax layer are used to train a k–nearest neighbour classifier (henceforth kNN), whose aim is to produce a fine–classification based on the activations of both borehole image data used. The output of this kNN classifier is the definitive lithofacies classification result considered when evaluating the net.

K–nearest neighbour classifiers belong to the non–parametric classifiers family. The classification of new data is achieved by computing the value of a cost function (which is commonly the euclidean distance) that depends on the values of each point of input data and the centroids or mesoids of each class in the features hyperspace. Each input point will be assigned to the class with the smaller associated cost. In our case, each class is associated with a certain lithofacies. For further reading on kNN classifiers see Altman (1992) and for some practical applications see Liao and Vemuri (2002) and Shen and Chou (2005).

These classifiers should not be confused with other common methods also based on the iterative computation of the data centroids or mesoids, such as k–means algorithm (Steinbach et al., 2000), which is an unsupervised segmentation technique.

5.3.3 Overall architecture

As introduced in Section 5.3.1, our structure is based on two CNNs, one for each type of borehole image data log. Each image sample, for each type of data, is introduced on its corresponding CNN, obtaining the features that are used as input to the kNN classifier which gives the definitive prediction of lithofacies. An illustration of the overall architecture with the complete diagram of the proposed method is shown in Figure 5.3. As it can be seen in this image, the results obtained right before and after the softmax layer of each CNN (ultrasonic data at the top and resistivity Data at the bottom), are concatenated and input into the kNN Classifier to obtain the lithofacies prediction.

5.3.4 Training subset division

The first step in the process is to divide the input data into subsamples. As lithofacies can be considered a discrete curve (on which each value per depth represents a categorical class), each depth measurement represents a single depth sample to be used for training or prediction. However, if we were to divide our input borehole images into single-row subsamples, the information contained on each row would not be sufficient to obtain consistent results because the geological structures we are trying to identify using the autoencoders are contained on bigger depth interval windows. This means that the geological information contained on single–row image subsamples would most likely not be enough to obtain consistent results, as the structures that are presumptuously linked to the property we are trying to predict (lithofacies) could not be contained on them. This is specially true in heterogeneous reservoirs like the ones found in the pre-salt carbonates.

In order to be able to capture as much geological information contained on each image subsample as possible, we propose to use a 21 pixel sliding window (which in this case is equivalent to a depth window interval of 10.67 *cm* or 4.2 inches) to crop the images, so that each subsample will include not only the information of that specific row, but also the information about its surroundings. Defining the optimal window size is an important task that could lead to better results and even a better understanding of how the information contained on borehole images is related to petrophysical and geological properties. Even though this process has not been implemented nor tested in this work, it will be addressed in future work on this area.

Once the images have been cropped using the same sliding window, and before any further process, it is essential to divide our data into training and testing subsets. In order to obtain robust results, we decided to divide our data into continuous subsamples of 100 lines each (this is equivalent to 50.8 *cm* or 20 inches). Trying to predict continous blocks is a more robust way to test the model rather than using randomly distributed single points, as it pushes the model to produce continously-consistent estimations the same way these models are expected to work in realistic oil/gas industry operations. The $193,307$ datasamples were then divided into training, validating and testing groups as follows:

1. $135,407$ (70.05%) as training samples on blocks with different samples each,
2. $28,900$ (13.95%) as validation samples on a total of 289 continuous blocks of 100 samples each,
3. $29,000$ (15.00%) as testing samples on a total of 290 continuous blocks of 100 samples each.

The first divisions of training, testing and validating subsamples, from 420 m to 653 m, are shown in Figure 5.4, where training intervals are identified as green regions, validating intervals as red regions and testing intervals as blue regions. It is important to notice that in order to prevent overfitting a certain number of samples N above and below each testing and validating intervals are not considered for training (left–out), which ensures that no sample used in the training of the net is used on its validation or testing procedures. The number of pixels to be left–out (N) depends on the size of the sizepad used to crop the input images SW and can be calculated by

$$N = \lfloor \frac{SW - 1}{2} \rfloor \tag{5.1}$$

.

5.3.5 Training technical information

Each CNN was trained using a gradient–descent algorithm[1] in order to find the optimal weights of the net so that the activations extracted from each type of data can be used for predicting the rock facies at each depth sample. The overall training process took over 3 days using a GeForce GTX 1080 GPU and Cuda Toolkit version 8. In this work we used MatConvNet library (Vedaldi and Lenc, 2015) for all NN architectures implemented, which is an open–source toolbox designed for implementing DNN in MATLAB®.

[1]For further information regarding the gradient–descent algorithm, see Baldi (1995); Vapnik (1999); Ruder (2016).

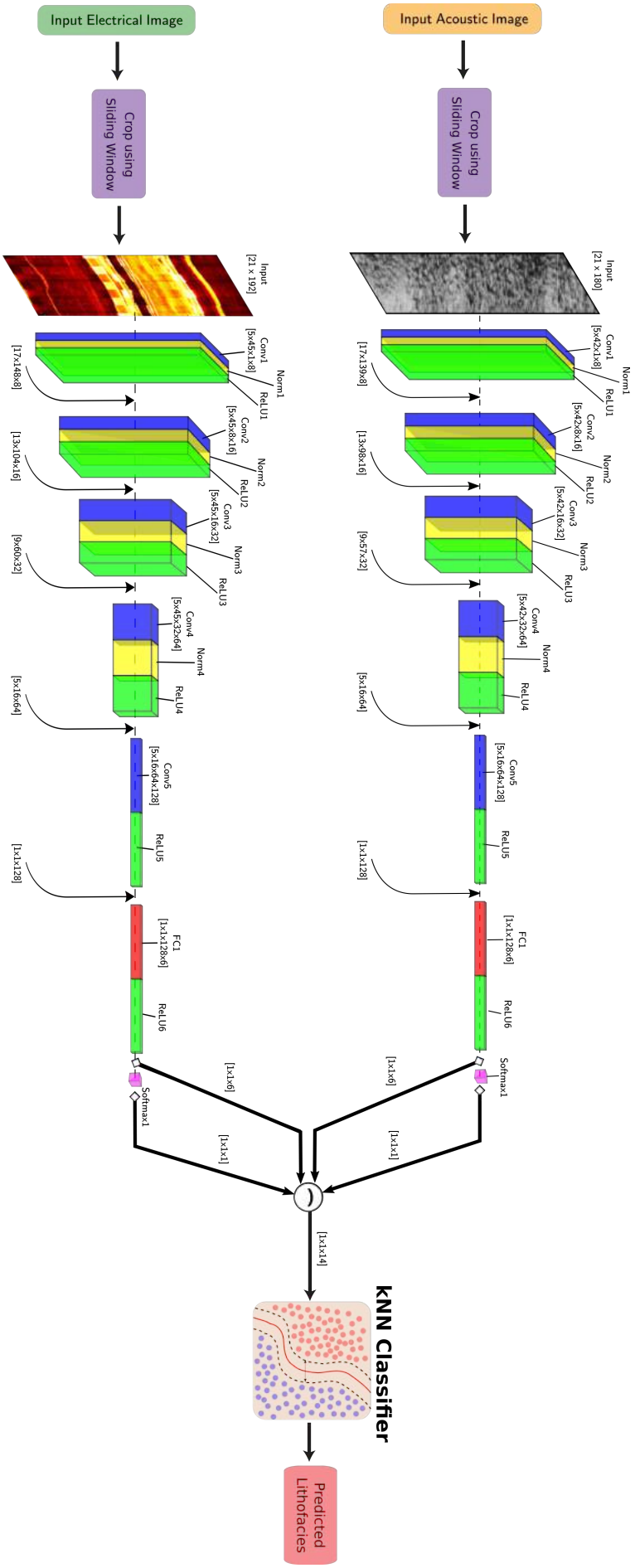


Figure 5.3 – Diagram of the complete workflow used to automatically predict lithofacies using ultrasonic and microresistivity borehole image logs. The first step of the process is to crop the input images using a sliding window. Then the cropped images go through the convolutional neural network of its specific type of data, producing the activations and probabilities of belonging to each type of lithofacies class. These probabilities and activations are then concatenated, for both types of data, and inputed into a k–nearest neighbour classifier, which returns the final lithofacies prediction.

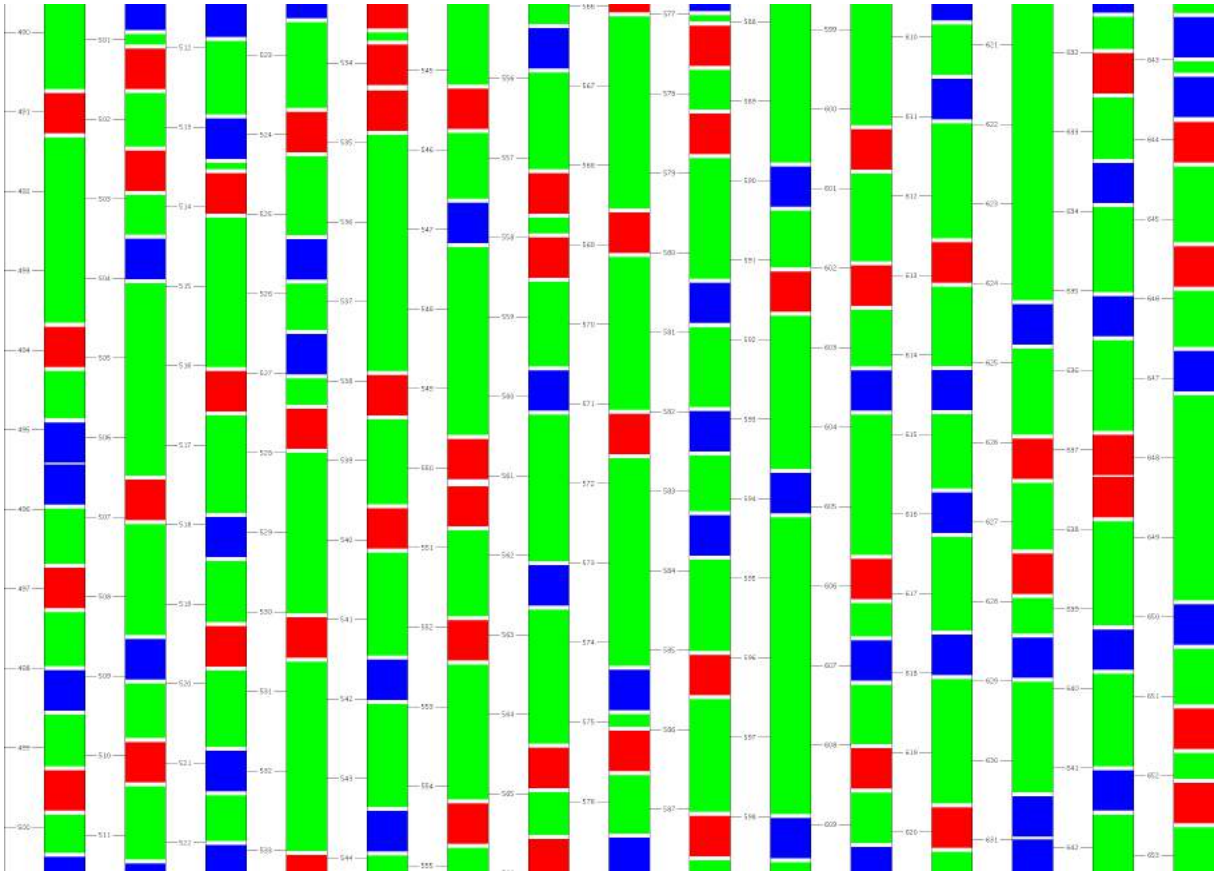


Figure 5.4 – Example of the first divions of subsamples into training (green regions), validating (red regions) and testing (blue regions). As it can be seen, validating and testing subsamples were selected to form continuous 100 pixels blocks, in order to obtain consistent and robust results. It is important to notice that blank gaps were left between testing/validating and training regions with a number N (5.1) of pixels above and below these regions to ensure that no sample used in the training of the net is used again on its validation or testing.

5.4 Lithofacies characterization results

The results shown in this Section are based only on the testing samples described in Section 5.2. No sample used during the training or validation process was used for testing purposes. At the same time, the results displayed in this Section were obtained for continuous 100 lines samples in an attempt to prove the robustness of the proposed method.

The overall classification success for all tested samples was 97.69%. In order to evaluate the accuracy of the classification we used two tools: a classification confusion table and the receiver operating curve (henceforth, ROC) plots of each lithofacies class. A confusion table shows the number of times that each target class was predicted as every other type of class, thus allowing us to visualize the classes that are most usually confused by the classifier, as well as the accuracy of prediction of each class.

The confusion table of the tested data is shown in Figure 5.5. As it can be seen there, even though the number of samples of each class was not uniform (ARGILITO and CALCILUTITO have far less samples than the rest of the classes) the results show that the predictions of classes appear to be consistent: no class was predicted with an accuracy inferior to 94%. The results seen in the confusion table also show that the pair of classes the classifier confuses the most is FOLHELHO–ARGILITO.

In order to evaluate how well defined each one of these classes is for the trained classifier, we plot the ROC for each class (Zweig and Campbell, 1993). These plots show how the rate of true positives (sensitivity) changes depending on the rate of false negatives (1−specificity). In plain words, these curves show how the success in classification changes in a binary classifier depending on the threshold chosen as condition of true prediction. The ROC plot for each Facies class for the proposed method applied to the testing data is shown in Figure 5.6. The dashed lines on these plots indicate a random guess: any curve below them (with an area under the curve below 0.5) is considered worse than a random guess and thus indicates that that certain class is not defined for the tested classifier. On the other hand, the higher the area under the ROC (the closer to 1) the better the classificator.

As it can be seen in Figure 5.6, all classes have an area under the curve above 0.95, which is much greater than 0.5 (random guess). It is possible to notice that the classes that are better defined for the trained classifier are: CALCILUTITO, DIABASIO and ARGILITO, while the worse defined one is FOLHELHO. This result is consistent with the previous results shown in the confusion table of Figure 5.5.

In Appendix B, from Figure B.1e to Figure B.59e we display the top 60 worse predicted blocks, sorted from worse (the whole block was misclassified) to better (the block samples were only partially misclassified). The first three columns display the Depth, Ultrasonic data and Resistivity data of that interval, while the forth and fifth columns show the

Figure 5.5 – Confusion Table obtained for the tested data and using the workflow proposed in this Section for lithofacies prediction using borehole image logs. Green cells indicate right predictions (where the class predicted by the classifier is the same as the original target class), red cells indicate wrong predictions (both false positives and true negatives), gray cells show statistics for each type of predicted or target classes and the blue cell on the bottom–right corner indicates the overall sucess. The text labels inside the green and red cells indicate the number of hits and their relative percentage with the overall total number of samples of all classes. The text labels inside the gray cells indicate the success rate in classification on top of their equivalent error rate (which they add up to 100%).

target facies of that interval and the prediction made by the model proposed in this Section.

An example of the evolution of the first 8 activations of each layer through both CNNs for the sample depth sample is shown in Figure 5.7, for the Ultrasonic data, and in Figure 5.8 for the Resistivity Data. As it can be seen there the activations of both nets capture different patterns of the inputed data. The activations of the Resistivity data seem, however, more heterogeneous than the activations obtained for the Ultrasonic data.

Lastly, an example of the 8 most proeminent weights of each ConvLayer of both CNN is shown in Figure 5.9. As it can be seen there, the weights of the first layer of the ultrasonic

CNN (top) seem to produce higher activations for bumpy data, which suggests that these layers look for porous formations, which might be correlated with some of the lithofacies classes further inside the net. On the other hand, the weights of the first layer of the Resistivity CNN (bottom) try to find more complex patterns such as fractures (second image on starting from the top on column f) or more gradative horizontal variations of resistivity. As it was expected (due to the increased complexity of each layer) the weights of deeper layers display more complex patterns.

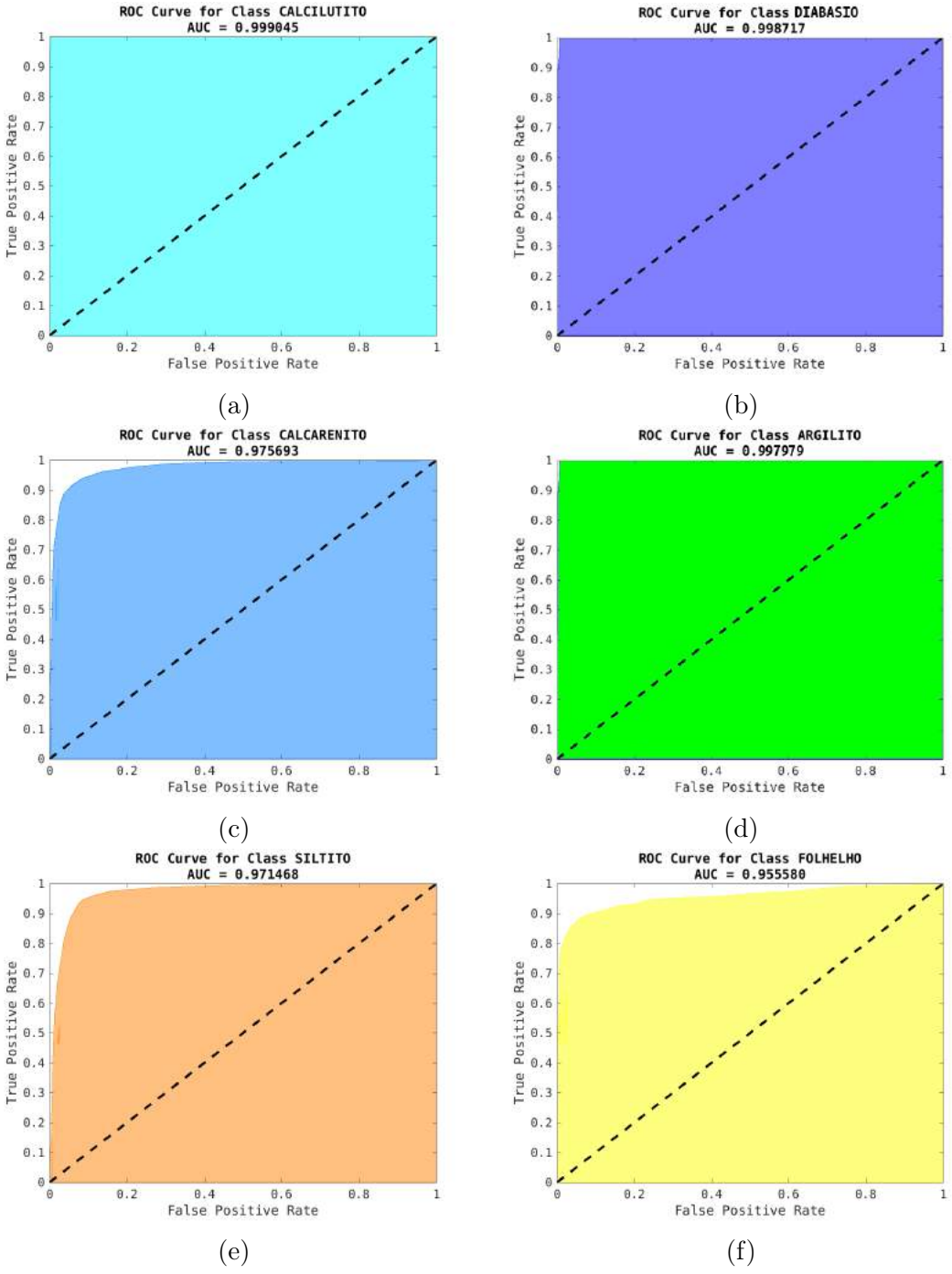


Figure 5.6 – *ROC Curves for each one of the lithofacies classes used in this work: a) CALCILUTITO; b) DIABASIO; c) CALCARENITO; d) ARGILITO; e) SILTITO and f) FOLHELHO. The dashed lines on each plot represent the equivalent curve of a random guess (untrained random classifier). As it can be seen, the area under each curve is much greater than the equivalent of a random guess (0.5) and very close 1, indicating that each class is almost perfectly defined for the trained classifier. These results are consistent with those shown in the confusion table shown in Figure 5.5.*

5.5 Conclusions

We trained two convolutional neural Networks with 17 layers each (combining convolutional, ReLU, fully–connected, softmax loss and batch normalization layers), with almost identical architectures, each one of them for a specific type of borehole image log (ultrasonic and resistivity) using the cropped borehole image data logs –cropped with a 21 pixel sliding window–.

Using this sliding window rather than just single–row vectors was considered necessary to preserve the geological structures present in the borehole image data. With the method proposed in this paper, these geological features remain intact, thus allowing the CNNs to extract information about the whole window and about the whole structure contained inside that window. This detail is specially important when trying to link borehole images with geological properties that highly depend on the values of the surroundings, such as the lithofacies we are trying to predict. To the best of our knowledge, all previous studies in the lithofacies classification field used single–point methods. In this work we have shown that using convolutional methods with borehole images seems to improve greatly the amount of information that we can extract from this type of data.

As we showed in Section 5.4, the predictions obtained with this model were competitive enough, achieving an overall success of 97.67%. These classifications were carried out using randomly centered continuous blocks with 100 lines each. We decided to use continuous blocks rather than single randomly located points in order to obtain more reliable and robust results, as well as to stress the model so that the results could be compared directly with more realistic production conditions.

These results are a clear indicator of the correlation between borehole image logs and geological/lithological properties. For several years this type of data has been seen by petrophysicists as a helping tool in the reservoir characterization process. However, the amount of information of borehole images to be taken into account in each case when analyzing a certain reservoir was more of a subjective feeling or an intuition than an objective measure. Our method shows that this correlation not only exists but that it can also be objectively quantified.

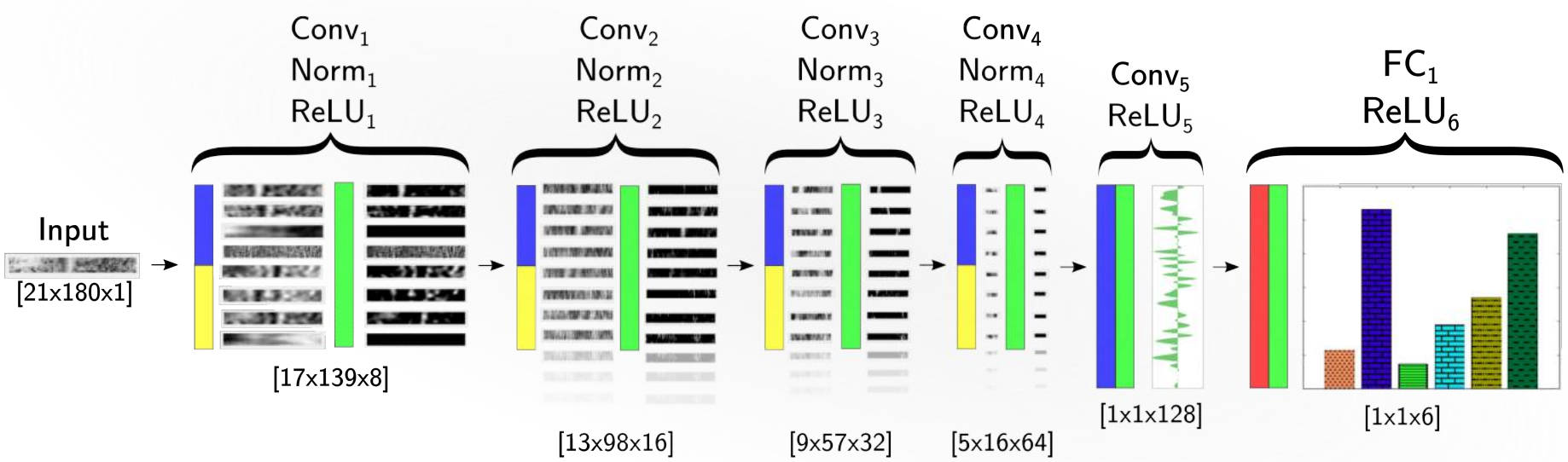


Figure 5.7 – Example of the first 8 activations of each layer in the ultrasonic CNN for a certain depth sample of UBI data. The input shown in the left is a sample of cropped UBI data (using the sliding window introduced before in this chapter, to preserve geological information of the surroundings). While this sample is processed through the net, different features are consecutively extracted from it. The first group formed by the Conv1 (blue) and Norm1 (yellow) layers takes the input cropped image and applies different filter kernels (feature extraction functions), thus enhancing different properties of the input data. After that, these filtered activation maps are processed by the ReLu1 (green) layer, which introduces non-linearity to the process. The following groups act in the same way to these maps, extracting more complex information from each subsample as more deeper in the net the layer is located. The final group, composed by FC1 (red) and ReLu6 (green) layers transform the weights of the previous layers into probabilites. Thus, the output of the net is a set of 6 probabilites, each indicating the probability of that cropped sample to belong to each class of lithofacies (see the legend shown in Figure 5.2). This illustration could be seen as an example of how the activation maps of a sample input image evolve through the trained net.

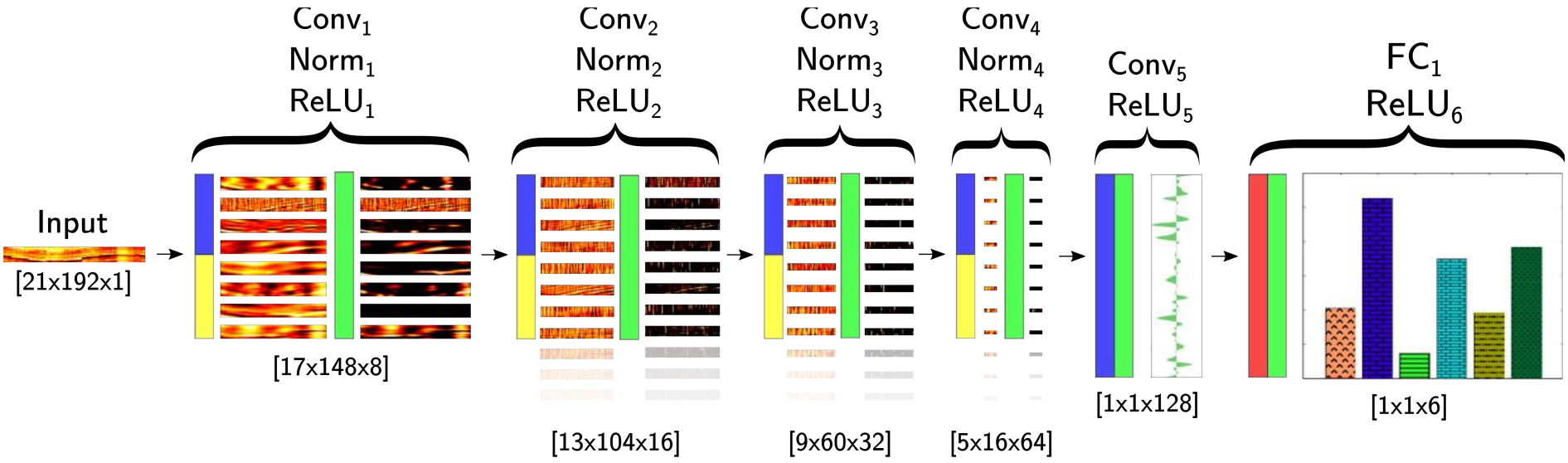


Figure 5.8 – Example of the first 8 activations of each layer in the microresistivity CNN for a certain depth sample of FMI data. The input shown in the left is a sample of cropped FMI data (using the sliding window introduced before in this chapter, to preserve geological information of the surroundings). While this sample is processed through the net, different features are consecutively extracted from it. The first group formed by the Conv1 (blue) and Norm1 (yellow) layers takes the input cropped image and applies different filter kernels (feature extraction functions), thus enhancing different properties of the input data. After that, these filtered activation maps are processed by the ReLu1 (green) layer, which introduces non-linearity to the process. The following groups act in the same way to these maps, extracting more complex information from each subsample as more deeper in the net the layer is located. The final group, composed by FC1 (red) and ReLu6 (green) layers transform the weights of the previous layers into probabilites. Thus, the output of the net is a set of 6 probabilites, each indicating the probability of that cropped sample to belong to each class of lithofacies (see the legend shown in Figure 5.2). This illustration could be seen as an example of how the activation maps of a sample input image evolve through the trained net.

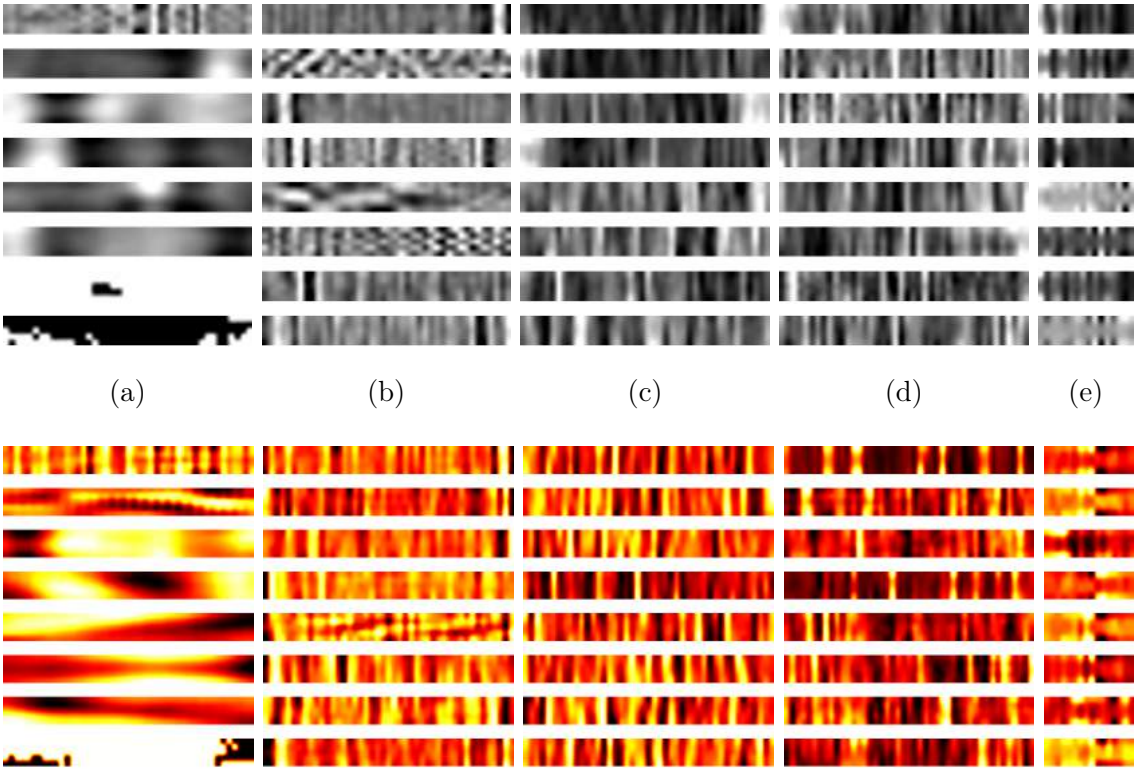


Figure 5.9 – Example of the 8 most proeminent weights of each ConvLayer of both CNN: (a) to (e) weights of convolutional layers from 1 to 5 of the UBI data CNN; (f) to (j) weights of convolutional layers from 1 to 5 of the FMI data CNN. The weights of the UBI CNN are shown on top, while the ones regarding the FMI CNN are on the bottom. Please, notice that these images do not match the sizes of the actual weights of the trained CNNs, but have rather been resized so that they could be displayed here. As it was introduced before, the weights of the first layers of the UBI CNN (a) seem to be sensitive to heterogeneity, which could be intuitively related to porous rocks in the borehole wall; while some weights of the first layers of the FMI CNN (f) seem to be sensitive to fracture-like formations (see the second weight map starting on the top of the (f) image).

Chapter 6

Estimation of Petrophysical Logs using Stacked Autoencoders and Borehole Image Data Logs

Estimating petrophysical properties from well-logs is a valid solution which might not only provide results with a reasonable level of accuracy (Mohaghegh et al., 1995; Perez et al., 2005; Hamada and Elshafei, 2010; Abdideh, 2012; Rafik and Kamel, 2016; Elkatatny et al., 2017; de Jesus et al., 2016), but also may help oil/gas industries reduce the total cost invested in reservoir characterization, for instance by using methodologies, like the one proposed in this section, to estimate petrophysical properties from borehole data.

In this context, we propose a novel method to use borehole image logs (borehole image data made available by PETROBRAS, see Section 2.6) and deep networks –Stacked autoencoders– to estimate effective porosity and permeability logs with an accuracy of over 96% for the tested data. This method allow us to extract information that encompasses a certain interval of the borehole image data logs, rather than just single points, as it was proposed in the previous work in the field (Mohaghegh et al., 1995; Perez et al., 2005; Hamada and Elshafei, 2010; Abdideh, 2012; Rafik and Kamel, 2016; Elkatatny et al., 2017; de Jesus et al., 2016). These intervals or windows are essential because they allow us to keep together some geological structures observed in the borehole wall, thus using the information of the surroundings of each depth and taking them into account in the final estimation of the petrophysical log. In this work, we discuss how this surrounding approach may present benefits when compared to using single row measurements.

It is worth mentioning that before any of this borehole image data can be used it is essential to process them in order to remove both the natural (such as natural breakouts or fractures) and artificial artifacts (like the ones caused by the drilling or measuring tool). The presence of these artifacts would alter the trained models, inducing error

in the estimation, as they do not have any geological meaning and therefore they do not contribute to the estimation of the petrophysical measurements on which we are interested here.

In order to adjust the ultrasonic borehole image data to these conditions, first the pad-removal filter designed by Valentin (2018), and presented in 3.4.2, was applied to the ultrasonic borehole image data, so that the artifacts caused by the pads of the logging tool would be removed and the real values of ultrasonic response of the rock lying underneath them would stand out. Afterwards, a contrast–limited adaptive histogram equalization (CLAHE) (Zuiderveld, 1994), as explained in 3.2, was applied to it in order to increase the local contrast of the image, as an attempt to enhance local geological features and structures. In this case, no other artifacts (such as breakouts or fractures) appeared on the data, so none of them were removed.

On the other side, the microresistivity borehole image data was set to logarithmic scale and afterwards the same CLAHE enhancement technique was used on it, in order to increase the local contrast. In this case no artifacts appeared on the Microresistivity image, so it was not necessary to remove them.

This chapter is divided as follows: in Section 4.3 and 6.2.1, we explain how AEs and SVM systems work and with what purpose they are normally used for. In Section 6.2.2, we explain in detail the novel method proposed in this Section, as well as its structure. In Section 6.3, we present the results obtained for all tested cases and data, comparing them with the results obtained in previous works in the field, also showing the best case estimations along with the original logs. Finally, in Section 6.4, we expose our conclusions about the results and the proposed method.

6.1 Previous work

The research for theoretical and empirical relationships between petrophysical magnitudes is not new. Archie et al. (1942) showed that electrical measurements could be used to estimate rock porosity. The empirical law that Archie established on his work made other authors attempt estimating permeability by trying to find empirical equations that could relate permeability to other physical properties such as porosity or water saturation (Leverett et al., 1941; Tixier, 1949; Timur et al., 1968; Jorgensen et al., 1988). Koponen et al. (1997) modified Kozeny-Carman equations of multiphase flow (Kozeny, 1927) and added effective porosity and tortuosity terms to it, as well as proved a very consistent relation between their predictions and experimental results. Even though these early empirical laws seem to provide accurate results, this accuracy seemed to be strongly dependent on the conditions and factors under which the tests were carried out.

More recently, some authors have proposed the use of well-logging data to estimate permeability and/or porosity (Mohaghegh et al., 1995; Perez et al., 2005; Hamada and

Elshafei, 2010; Abdideh, 2012; Rafik and Kamel, 2016; Elkatatny et al., 2017). Instead of trying to find an analytical or empirical equation between well-logs and petrophysical properties, these techniques take advantage of modern computational methods and use artificial intelligence to obtain these experimental relations.

Borehole image logs have been used for porosity and permeability estimation based on the estimation of fracture aperture (Cheung et al., 1990; Luthi and Souhaité, 1990). Using a similar approach, de Jesus et al. (2016) proposed to use borehole ultrasonic image logs (Williamson et al., 1999) to estimate matrix permeability by obtaining a segmented version of the ultrasonic impedance borehole image. Even though their method achieved good results for the tested cases, the segmentation process –on which the final result relies greatly– requires the user to have a certain degree of geology knowledge and experience in borehole image analysis. On the other hand, Li and Misra (2017) used variational autoencoders to generate in-situ pore size distribution, which can be further processed to obtain residual saturation and permeability.

In this chapter, we make use of the relationship between borehole images and petrophysical properties exploited by de Jesus et al. (2016); Cheung et al. (1990); Luthi and Souhaité (1990) and use it, along with deep learning artificial intelligence methods, to estimate permeability and porosity with a high degree of accuracy.

6.2 Methodology

Autoencoders cannot be used in regression problems directly, as they are strictly unsupervised networks. In order to make regression or classification tasks possible it is necessary to add an extra structure after the autoencoders to link the encoded features extracted by them to the parameters we wish to estimate. In our case, a Support Vector Machine (henceforth, SVM) was trained using the encoded features obtained from the borehole image log samples to predict porosity and permeability data logs.

6.2.1 SVM regressors

Support vector machine regressors belong to the family of parametric non-probabilistic binary regressors. These models aim to construct a vector on a feature hyperplane such as the distance between the different samples and the vector itself is maximized. The set of vectors that maximized the distance between the clusters in the created hyperplanes –called Support Vectors– are used as boundary limits for further predictions. Any new point will be allocated at one side of the vector, and so its predicted value will depend on a certain loss-function and its distance to these borders in the hyperplane space. For further information on SVM classifiers and regressors see Tong and Koller (2001); Joachims (1998); Inglada (2007).

6.2.2 Proposed method

In this work, we propose a novel workflow to estimate formation porosity and permeability logs using a set of 3 level Stacked AE for each type of borehole image data (one for the ultrasonic and other for the microresistivity data), which are used to encode the information contained inside these Borehole Image Logs, and use the encoded information of each image in an SVM Classifier trained to estimate permeability and porosity data logs. The internal structure of each one of these AE was composed, respectively, by: 200[1], 100 and 50 hidden neurons. An example of this hidden features extraction, for a sample borehole ultrasonic image, using the first autoencoder (AE_1), can be seen in Figure 4.3b, where a sample of 4 internal encoded logs are shown. As it can be seen in this Figure, each one of the extracted textural features captures a unique property of the input image.

The first step in the process is to divide the input data into subsamples. As both the permeability and porosity measurements are formed by single-points measurements, each depth measurement represents a single sample to be used for training or prediction. On the other hand, we used a 21 pixel sliding window (equivalent to a depth window interval of 5.33 *cm* or 2.1 inches) to crop the borehole data images, so that each subsample will include not only the information of that specific row, but also the information regarding its surroundings. The importance of using sliding windows when dealing with geological image data in order to preserve crucial geological structures was already discussed in Section 5.3.4.

Once the images have been cropped using the same sliding window, and before any further process, it is essential to divide our data into training, validating and testing subsets. In Section 5.3.4 we already discussed the importance of predicting continuous blocks of petrophysical properties instead of single randomly located samples in order to obtain robust results. We decided to divide our testing and validating data into continuous subsamples of 300 lines each (this is equivalent to intervals of 76.2 *cm* or 30 inches). Then we divided our data into 80% training ($15,110$ samples) and 20% validating ($3,777$ samples) groups.

After that, two different three–level stacked AEs are trained using the training subset, one for each one of the cropped images –one for the cropped ultrasonic borehole image and another one for the cropped microresistivity borehole image–.

Once the stacked AEs have been trained, the input borehole images are encoded using them, and this encoded information is stacked and prepared to be used by the SVM regressor. This process, which is repeated for both the ulrasonic and microresistivity borehole images, was illustrated in 4.3b.

[1]Using 200 hidden neurons on the input autoencoder (AE_1) net represents compressing the data present in the borehole images using a compression factor of 95%. In other words, this can be seen as forcing the net to learn to represent the borehole images using 95% less amount of information than the original images.

Afterwards, the extracted encoded features, of both the ultrasonic and microresistivity borehole images, are introduced into the SVM structure, along with the permeability or porosity training samples. During this process the SVM is trained to learn the internal model that links these petrophysical properties and the features extracted from the borehole images.

The stacked AEs and SVM are put together into a single architecture so that the testing and estimation procedure that follows consists simply on introducing the Borehole Images into the stacked network. At this point it is possible to calculate the accuracy of our method by using the testing subset data. In our work we have used two different coefficients to quantify the accuracy of our method:

1. Coefficient of determination (R^2) between predicted and original testing data. This coefficient quantifies the linear correlation between two variables (Benesty et al., 2009). In this work we use this coefficient as a measure of the quality of the estimation. The closer this value is to 100%, the better the model is at estimating the petrophysical parameters.

2. Normalized Root Mean Squared Error ($\%RMSE$)[2] between predicted and original testing data. This coefficient is a measure of the sample standard deviation of the differences between the predicted and original data (Willmott, 1981), normalized by the total range of the original data. The lowest this value is, the better the model is at estimating the petrophysical parameters.

A diagram of the complete workflow proposed in this work is shown in Figure 6.1.

6.2.3 Formation permeability and effective porosity measures

The permeability and porosity measurements used in this paper were obtained applying Timur-Coates models to NMR data (Timur et al., 1968). Even though these measurements are not direct porosity/permeability measurements, it has been shown over the years that NMR extrapolated petrophysical data presents a high correlation with real data (Mohaghegh et al., 1997; Li and Misra, 2017).

Figure 6.2 shows the histogram of both the petrophysical logs used in this work: the base 10 logarithm of the permeability measurements (left) and the effective porosity measurements in percentage (right). Extracting an interpretation regarding possible groups of reservoir rocks or characteristics from the information contained inside these histograms is a very complex task.

[2] The normalized root mean squared error can be obtained by dividing the Root Mean Square Error (RMSE) by the total range of the original data to be fitted. This measurement was preferred in this work because it is dimensionless.

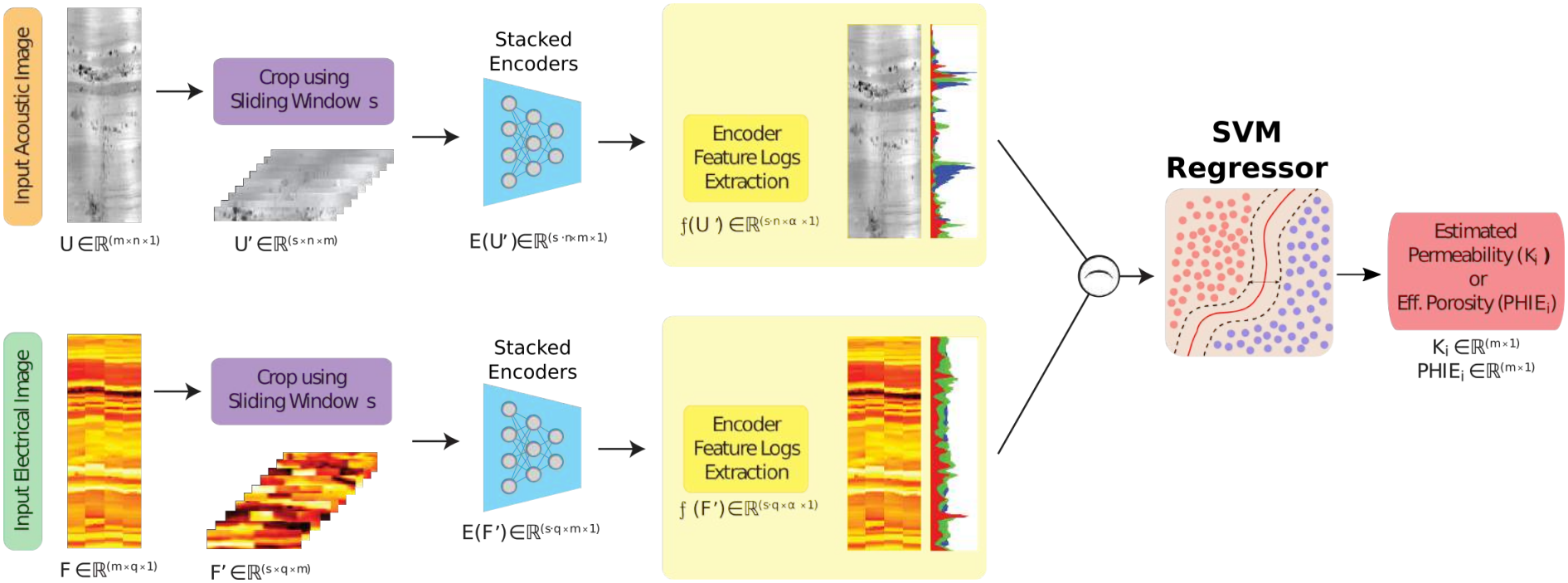


Figure 6.1 – Complete diagram of the workflow proposed in this paper to estimate petrophysical properties from borehole image logs. First the input borehole images are cropped using a sliding window of $[21 \times 180]$. The cropped images are separated into training and validating subsets. The training cropped images are used to train a 3–level stacked autoencoder, one for each type of borehole image log. Afterwards, the validating cropped images are encoded using these autoencoders. The encoded information is later used in the SVM regressor to predict the values of permeability and porosity for each cropped image. In the diagram it is possible to see how a sample UBI (top, left) and FMI (bottom, left) images are cropped using the sliding window mentioned before, and how each one of these cropped images are encoded using the AEs to extract the feature logs (encoded logs).

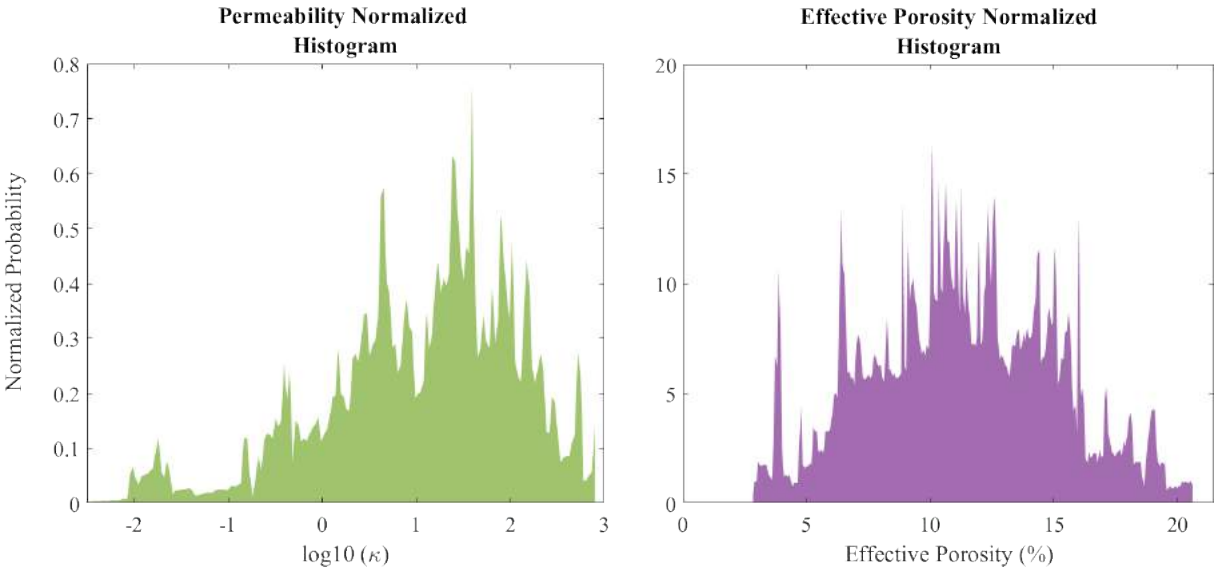


Figure 6.2 – Normalized distributions of base 10 logarithm of permeability measurements (left) and effective porosity measurements (right) of the petrophysical data used in this work.

6.3 Estimation Results

In this section, we present the results obtained when using the proposed workflow to estimate porosity –Section 6.3.1– and permeability –Section 6.3.2–. Table 6 shows the accuracy for the proposed method when estimating both the effective porosity and the formation permeability logs. These results will be discussed in the following sections.

Table 6 – Estimation of effective porosity and formation permeability measurements results. Estimation accuracy is quantified using the coefficient of determination R^2 and the normalized root mean squared error $\%RMSE^2$ between original and predicted effective porosity values.

	Effective porosity	Formation permeability
%RMSE^2 (%)	7.30	5.51
R^2 (%)	96.30	96.06

Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7 and Figure 6.8 display the data used in this paper, along with the original and estimated values for each petrophysical log: effective porosity and permeability. Due to the fact that the data covers thousands of depth samples it was necessary to split them in 5 different intervals in order to show them on these figures. In all of these figures the data is presented using the same structure: at the left (a) the dynamically normalized ultrasonic borehole image data is presented, next to it (b) the dynamically normalized microresistivity borehole image data is presented, followed by the effective porosity data (c) and the formation permeability data (d), respectively. Both curves use the same color code for displaying the results. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.

6.3.1 Effective porosity estimation results

As it can be seen in Table 6, the Normalized RMSE^2 value when estimating the Effective Porosity log is below 10% (7.30%), while the R^2 coefficient is fairly high (96.30%). Both these coefficient values indicate a high-quality estimation of the tested effective porosity log. The complete original log and these estimations is shown in Figure 6.4c Figure 6.5c, Figure 6.6c, Figure6.7c and Figure 6.8c .

Table 7 compares our best result in estimating effective porosity logs with previous studies in the porosity estimation field. As it can be seen our result is not only higher

Table 7 – Comparison between our best result for the estimation of effective porosity measurements the results of previous authors.

	R^2(%)	Data used	Method
Hamada and Elshafei (2010)	66.00	RHOB[8]; GR[9]; NPHI;[11]DRT/SRT;[12]NMR(T2)[13]	FFNN[10]
Our method	96.30	UBI/OBMI[14]	sAE[15] + SV

[8] RHOB: Bulk Density Log.
[9] GR: Gamma Ray Log.
[10] FFNN: Multilayer Feed-forward Neural Network.
[11] NPHI: Neutron Porosity Log.
[12] DRT: Deep Resistivity Log / SRT: Shallow Resistivity Log.
[13] NMR(T2): NMR Decay Inversion Time T2 Log.
[14] UBI: Ultrasonic Borehole Image / OBMI: Oil-based Borehole Micro-resistivity Imager.
[15] sAE: Stacked Autoencoders.
[16] SVM: Support Vector Machine.

than the one that Hamada and Elshafei (2010) obtained, but it is very close to 100%, meaning that the estimated values are extremely close to the reference porosity log.

Lastly, Figure 6.3a and Figure 6.3b show an example of the internal encoding weights of the trained autoencoders for the fifteen first inner layers with the highest correlation with the effective porosity logs, for the (a) ultrasonic image data and the (b) microresistivity image data, respectively. It is worth mentioning that even though these filters show a great diversity, it is possible to notice that most of them seem to show higher activations for images with high variation in their grayscale tone for the Ultrasonic Borehole Data case (Fig 6.3a). This means that rocks that appear in ultrasonic images with high heterogeneity (i.e., due to all intrinsic heterogeneities and roughness of the formation) are more likely to activate the autoencoder of the ultrasonic data. This result supports the notion that porosity is somehow directly related to the ultrasonic variations displayed in Borehole Data, but in a more complex way than the naked eye can see.

6.3.2 Permeability estimation results

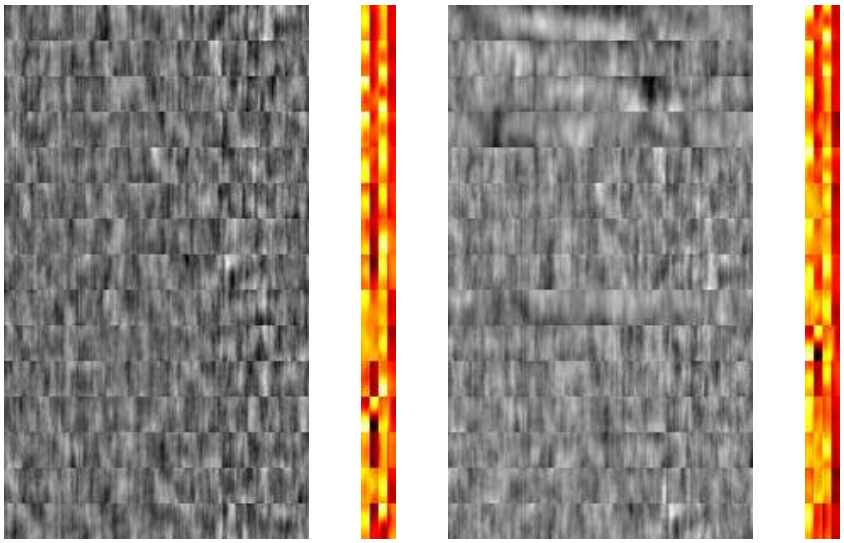
As it can be seen in Table 6, the normalized RMSE[2] value is below 10%, while the R^2 coefficient is slightly over 96%, indicating a high-quality estimation, similarly to the porosity estimation case. The complete original log and these estimations is shown in Figure 6.4d Figure 6.5d, Figure 6.6d, Figure6.7d and Figure 6.8d .

Table 8 compares our best result in estimating permeability logs with previous studies. As it can be seen our result is higher than all of them but the result obtained by Mohaghegh et al. (1995). However, it is worth noticing that the result Mohaghegh et al. (1995) presented on their work was tested on a total of 23 sample points, while our method was tested using over 3700 sample points: a considerable increase in the amount of data points. On the other side, our best result has a R^2 very close to 100%, meaning that the estimated values are extremely close to the reference permeability log.

Table 8 – Comparison between our best result for the estimation of permeability measurements and the results of previous authors.

	R^2(%)	Data used	Method
Mohaghegh et al. (1995)	96.30	Depth, GR[9], RHOB[8], DRT[12], ZSS[18]	ANN[17]
Hamada and Elshafei (2010)	93.75	RHOB[8], GR[9], NPHI[11], DRT/SRT[12], NMR(T2)[13]	FFNN[10]
Abdideh (2012)	89.00	Depth, CDT[19], DT[20], NPHI[11], RHOB[8], PHIT[21], WS[22], GR[9]	FFNN[10]
Elkatatny et al. (2017)	95.00	RHOB[8], GR[9], NPHI[11], DT[20], DRT/SRT[12], WS[22], Caliper, PHIT[21]	BNN[23] + ANFIS[24] SVM[16]
Our method	96.06	UBI/OBMI[14]	sAE[15] + SVM[16]

[17] ANN: Artificial Neural Network.
[18] ZSS: Zonal Subdivision Specification.
[19] CDT: Conductivity Logs.
[20] DT: Sonic Response Log.
[21] PHIT: Total Porosity Log.
[22] WS: Water Saturation Log.
[23] BNN: Backpropagation Neural Network.
[24] ANFIS: Adaptative Neuro Fuzzy Interference System.



(a) (b) (c) (d)

Figure 6.3 – First 15 internal encoding weights of the trained autoencoders that produce the encoded features with the higher correlation with the effective porosity for the (a) ultrasonic borehole data and (b) microresistivity borehole data; and with the higher correlation with the formation permeability log for the (c) ultrasonic borehole data and (d) microresistivity borehole data. Bright areas indicate high activation of neurons. The difference and variance between filters in this image show the complexity of extracting textural information from borehole images that might be linked to petrophysical data.

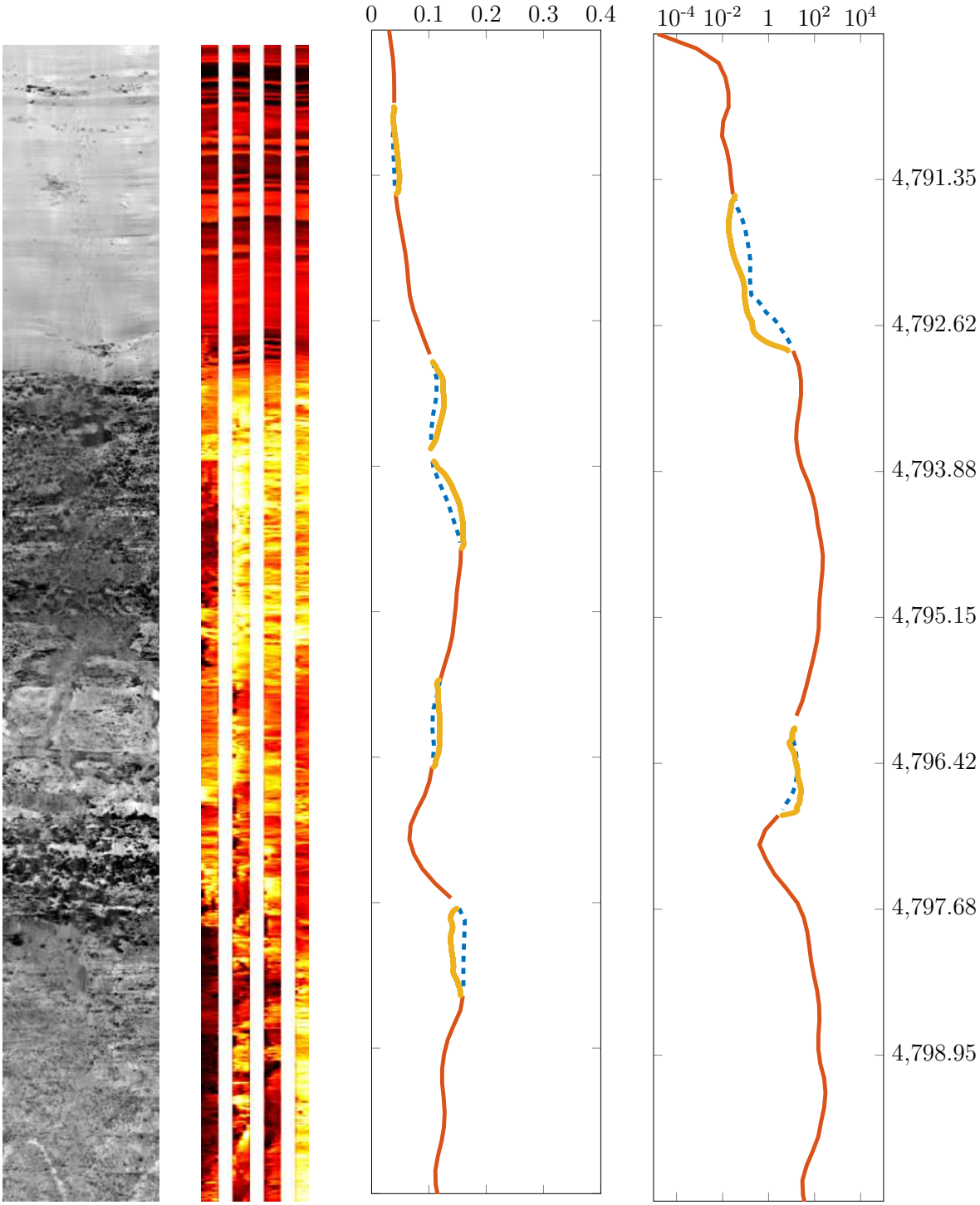
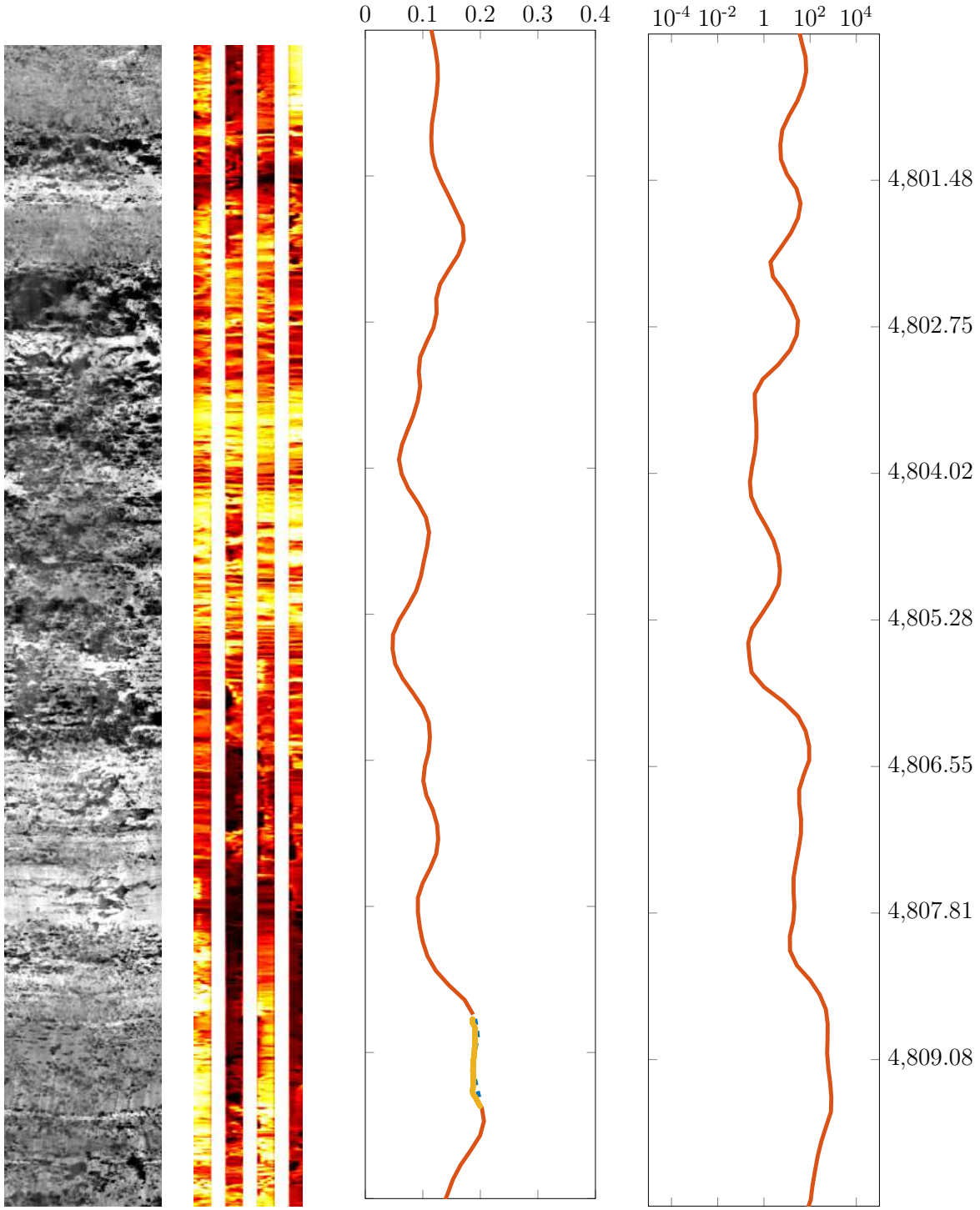


Figure 6.4 – a) Dynamically normalized ultrasonic borehole image used for the petrophysical parameters estimation test from depth 4790.084 to 4800.242 meters. b) Dynamically normalized microresistivity image used for the petrophysical parameters estimation test from depth depth 4790.084 to 4800.242 meters. c) Effective porosity logs and d) formation permeability logs predicted from borehole image data from depth depth 4790.084 to 4800.242 meters. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.



(a) UBI (b) FMI (c) PHIE (d) PERM

Figure 6.5 – a) Dynamically normalized ultrasonic borehole image used for the petrophysical parameters estimation test from depth 4800.244 to 4810.402 meters. b) Dynamically normalized microresistivity image used for the petrophysical parameters estimation test from depth 4800.244 to 4810.402 meters. c) Effective porosity logs and d) formation permeability logs predicted from borehole image data from depth 4800.244 to 4810.402 meters. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.

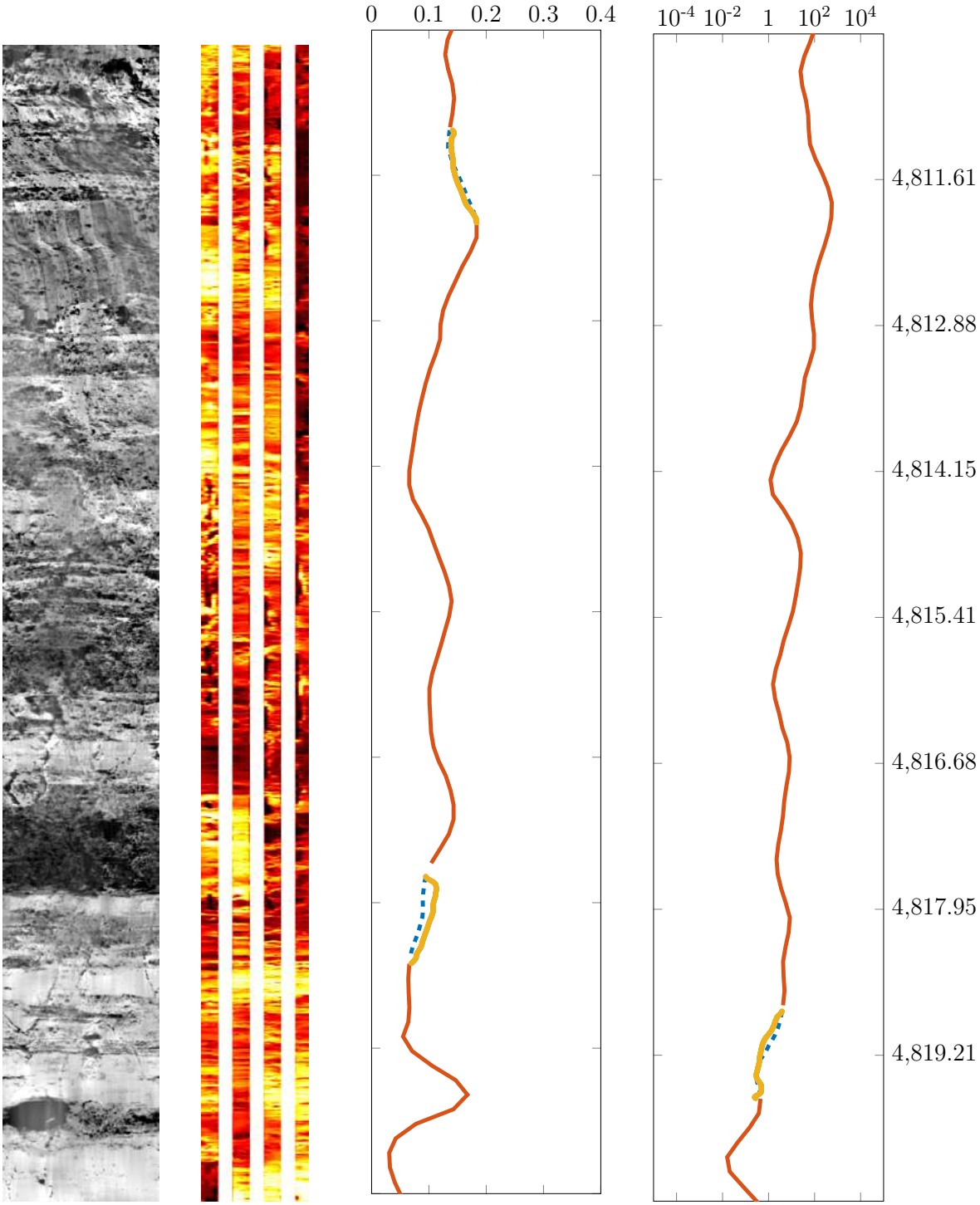
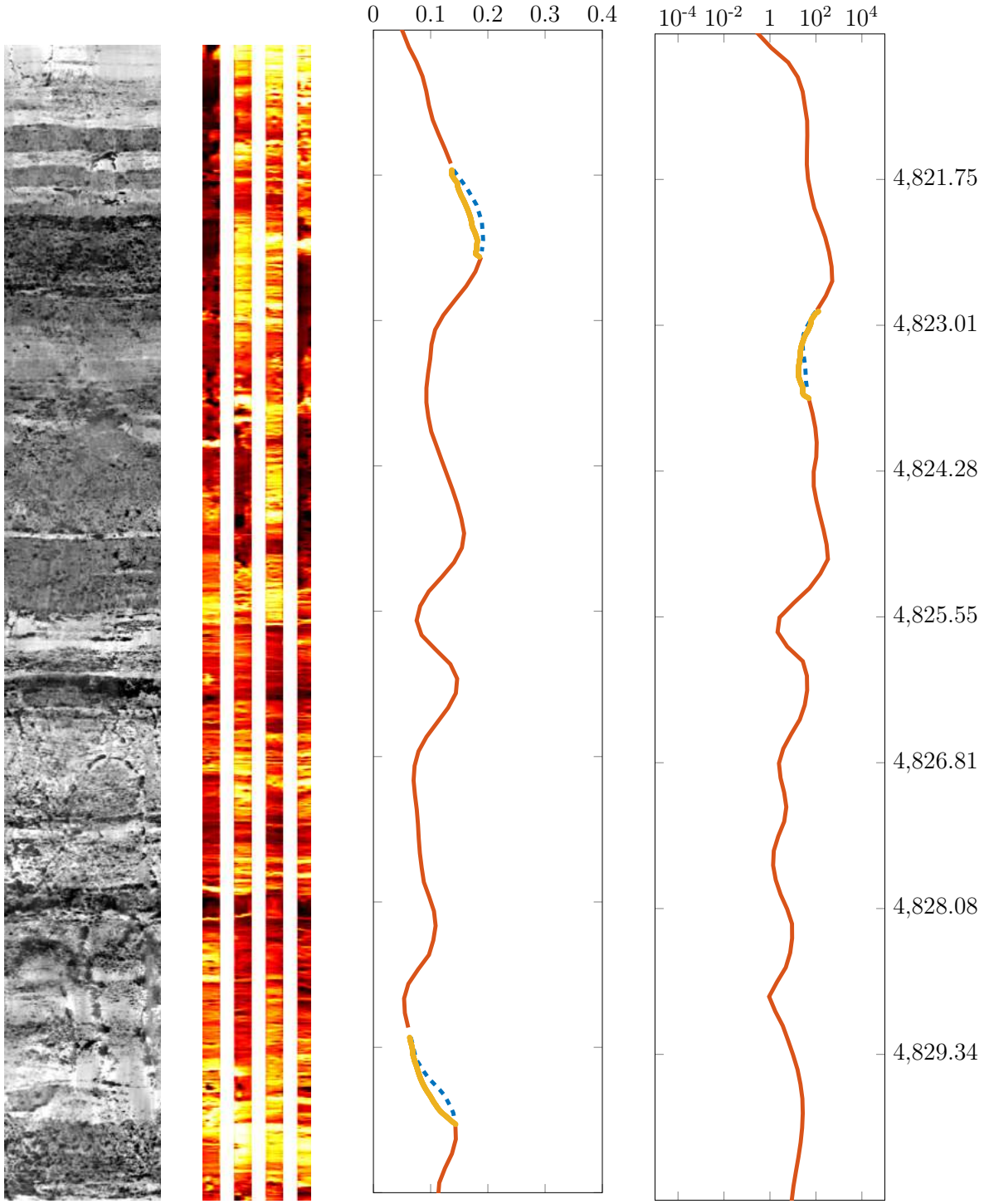
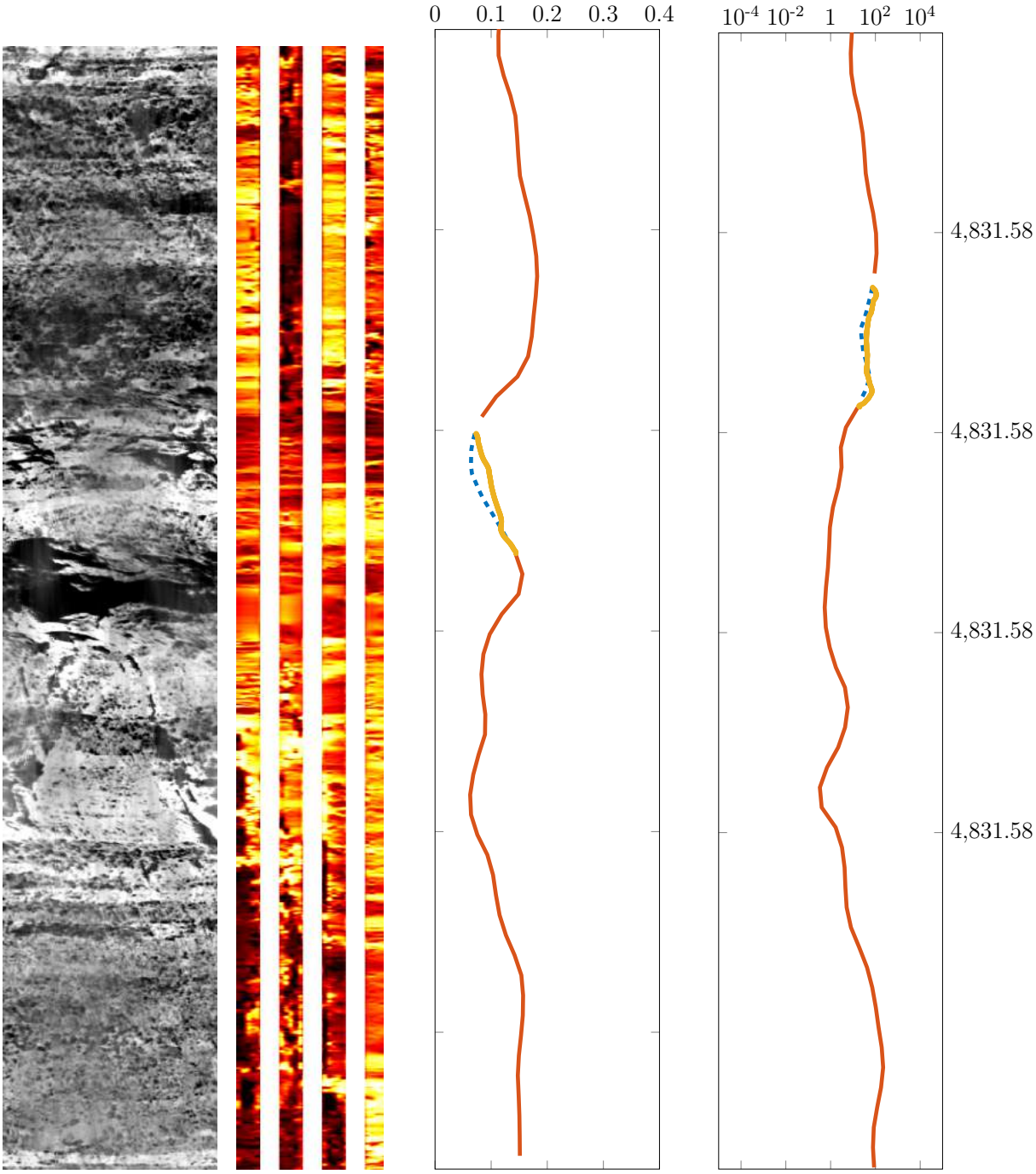


Figure 6.6 – a) Dynamically normalized ultrasonic borehole image used for the petrophysical parameters estimation test from depth 4810.404 to 4820.562 meters. b) Dynamically normalized microresistivity image used for the petrophysical parameters estimation test from depth 4810.404 to 4820.562 meters. c) Effective porosity logs and d) formation permeability logs predicted from borehole image data from depth 4810.404 to 4820.562 meters. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.



(a) UBI (b) FMI (c) PHIE (d) PERM

Figure 6.7 – a) Dynamically normalized ultrasonic borehole image used for the petrophysical parameters estimation test from depth 4820.564 to 4830.722 meters. b) Dynamically normalized microresistivity image used for the petrophysical parameters estimation test from depth 4820.564 to 4830.722 meters. c) Effective porosity logs and d) formation permeability logs predicted from borehole image data from depth 4820.564 to 4830.722 meters. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.



(a) UBI (b) FMI (c) PHIE (d) PERM

Figure 6.8 – a) Dynamically normalized ultrasonic borehole image used for the petrophysical parameters estimation test from depth 4830.725 to 4837.938 meters. b) Dynamically normalized microresistivity image used for the petrophysical parameters estimation test from depth 4830.725 to 4837.938 meters. c) Effective porosity logs and d) formation permeability logs predicted from borehole image data from depth 4830.725 to 4837.938 meters. The original values used for training the net are plotted in solid red lines, the original values used for testing the net are plotted in dashed blue lines, while the values predicted by the net are represented by the solid yellow lines. The original values used for testing the net were not used during the training procedure. The accuracy of the estimation was obtained by comparing the values represented by the blue dashed and the yellow lines.

6.4 Conclusions

In this chapter, we have described a novel method to estimate permeability and effective porosity logs using borehole image data logs and deep networks. By doing so we expected to show that ultrasonic and microresistivity borehole images contain inside them far more information than previously suspected, and that this information can be used for the estimation of these petrophysical property logs.

We used the cropped borehole image data logs –using a 21 pixel sliding window– and trained a 3–level stacked autoencoder with them. Even though this has not been addressed in this work, we plan to explore how the size of this window affects the final estimation and how to find an optimal size for each kind of application, depending on the type of studied reservoir and its expected geological formations, in future investigations. We used these cropped images to train a series of stacked autoencoders (each for each type of borehole image), which we later used to characterize the differences in the information of the reservoir contained within these Image logs, extracting their inner features and using them in a Support vector machine to predict either the effective porosity or permeability logs final values.

Using this sliding window rather than just single–row vectors was considered necessary to preserve the geological structures present in the borehole image data. With the method proposed in this paper, these geological features remain intact, thus allowing the autoencoders to extract information about the whole window and about the whole structure contained inside that window. This detail is specially important when trying to link borehole images with petrophysical properties that highly depend on the values of the surroundings, such as the properties our method estimates (porosity and permeability). All previous studies in the petrophysical estimation field used single–point methods. In this work we suggest that extracting the information of a certain window from borehole images seems to improve greatly the amount of information that we can extract from this type of data.

As we showed in Section 6.3, we obtained competitive results for both porosity and permeability estimations: $R^2 = 96.30\%$ and $R^2 = 96.06\%$, respectively. These estimations were carried on using randomly centered continuous blocks with 300 samples of width each. We decided to use continuous blocks rather than single randomly located points in order to obtain more reliable and robust results, as well as to stress the network so that the results could be compared directly with more realistic production conditions.

These results are a clear indicator that the correlation between borehole image logs and some petrophysical logs (at least the ones we tested in this paper) is a reality. For several years this type of data has been seen by petrophysicists as a helping tool in the reservoir characterization process. However, the amount of information of borehole images to be taken into account in each case when analyzing a certain reservoir was more of a

subjective feeling or an intuition than an objective measure. Our method shows that this correlation not only exists but that it can also be objectively quantified.

Our model performed better in estimating these petrophysical parameters than almost all of benchmark studied, on which our work was based. Even though Mohaghegh et al. Mohaghegh et al. (1995) had a slightly better result in the permeability prediction case ($R^2 = 0.24\%$ better than our method), it is important to notice that in their work they used far less testing points than us: they tested their method on a total of 23 sample points, while we tested our method on over 3700 sample points.

Taking into account the results we presented on this work and the current state of the art of petrophysical logs estimation we consider that this novel method might be considered an interesting approach for producing reliable estimations for both effective and permeability logs when using borehole image data logs. Apart from that, it is also worth mentioning that this workflow does not require the user to have any previous advanced knowledge regarding deep networks nor artificial intelligence nor geology in order to use it.

Chapter 7

Conclusions

The objective of this dissertation was two–fold: first, to evidentiate how the use of typical image processing techniques could help improve the quality of borehole image logs, enhancing some of their features and/or retrieving part of the information contained inside them; and second, to present the applicability of deep learning techniques to this kind of data, and how them –which have revolutionized the machine learning field– might be benefitial for the oil/gas industry.

In order to do so, we first introduced the different types of reservoir data that any geologist, petrophysicist or analyst has to deal with when trying to characterize a reservoir, in Section 2. Similarly, we introduced the concept of Deep Learning in Section 4, its advantages and the different techniques and architectures that have gained popularity over the last years, as well as how they could be applied to reservoir data.

Along this work we introduced 5 different novel methods to enhance, improve, characterize or extract information from borehole image logs. Three of these methods were based on shallow (non–deep) image processing techniques, which had the objective of apply some type of transformations (linear or not) to the images, so that some part of the information contained in them would be enhanced or removed: automatic breakouts detection from borehole image logs, Eccentricity correction on borehole image logs and automatic pad artifact removal on borehole image logs. On the other side, the rest of the methods were based on deep learning techniques, which had the objective to extract inner hidden features of the borehole images that could be later link to other type of petrophysical properties: estimation of petrophysical logs from borehole log image data and lithological reservoir characterization using deep convolutional neural networks.

7.1 Method for automatic breakouts detection from borehole image logs

In Section 3.3, we proposed a technique to automatically detect breakout structures from borehole image logs, based on the information of both the ultrasonic and transit–time data logs. The method requires the application of a standard–deviation filter (and, optionally, a gaussian filter) on the transit–time data and the dilation and segmentation of the resulting data. The geometric properties (such as area, centroid, etc.) from the segmented formations are extracted and used to find the structures that better match the conditions to be considered as breakout–pairs candidates.

Previous work on automatic detection of breakouts (Wessling and Pei, 2015) required introducing a specialized tool into the borehole, after drilling. It is important to notice that the total cost of any borehole drilling procedure is directly proportional to the time spent on the operation, so that introducing any additional tool to detect breakouts (apart from the conventional logging tools mentioned in Section 2 required for lithology characterization) will inevitably cause an increase in the final cost of the operation.

The method we presented, on the other hand, does not require any special tool to be introduced into the wellbore, as it is based on borehole image logs (which are already obtained in conventional drilling procedures and used for several purposes, as introduced in this dissertation). Apart from that, our method has presented a high accuracy in characterizing the geometry of these formations. The introduction of this technique in the current workflow of reservoir characterization and borehole image analysis could also be very simple.

7.2 Method for eccentricity correction in borehole image logs

In Section 3.4.1, we presented a novel technique to transform ultrasonic borehole image data logs in order to compensate the effect that irregular borehole shape and probe uncentering cause on the information contained in them. This effect is critical because it causes differences in the contrast of the borehole image that have nothing to do with the lithology of the wellbore wall (on which specialists are usually interested on).

Our method, which is an extension of Menger et al. (1994) work, uses transit–time image logs to obtain an estimation of a distance map between the probe and the borehole, which is later used on a linear transformation applied to the ultrasonic data to correct each sample point, deleting the effect that this distance had on the ultrasonic impedance value of that same sample.

Even though our method is semi–automatic (as it depends on a parameter to be regulated by the user), the results displayed in Section 3.4.1 show that our technique effectively increases the local contrast of the ultrasonic image, thus enhancing the lithological infor-

mation contained inside it and allowing some hidden structures from the Wellbore to appear.

7.3 Method for automatic pad artifact removal from borehole image logs

In Section 3.4.2, a novel method for removing pad artifacts from ultrasonic image logs was introduced. These artifacts are left by other types of logging tools, introduced before into the Borehole, which require or come with some type of centering mechanism (usually some type of pads or rollers) when making contact with the borehole wall. These artifacts have a great impact on the final Borehole Image, as they mask the lithological information we are aiming to capture with these probes (i.e., the acoustic or electrical impedance of the borehole wall).

Our method is based on the application of a specific pad–reject function to the ultrasonic data, so that the resulting image contains only the lithological information of the wellbore that was not caused by the artifacts. As it can be seen in the previously mentioned section, our methods shows very interesting results when applied to real borehole image logs, removing the pad–artifacts and enhancing only the lithological underneath. The results suggest a relvant gain in borehole processing methods, and it might help geologists and petrophysicists to analyze and exract information from these borehole logs more easily. To the best of our knowledge there are no previous methods nor techniques (publically published) regarding this topic.

7.4 Method for reservoir lithofacies characterization using deep convolutional neural networks and borehole image data logs

In Section 5, we described a novel method to automatically classify reservoir lithofacies using borehole image data logs and deep convolutional neural networks. By doing so we expected to show, the same way we did in Section 6, that ultrasonic and microresistivity borehole images contain inside them far more information than was expected, and that this information can be used for such a challenging and important task as the classification of lithofacies.

Our method was based on two identical convolutional neural networks with a total of 17 layers (5 convolutional, 4 normalization, 6 activation, 1 fully–connected and 1 softmax), one for each type of borehole image data log used (ultrasonic and microresistivity), whose function is to extract the inner high–order features of these image Logs. After training them, these nets (as explained in Section 5.4) automatically look for high–order features such as fractures or bumpy regions in the image logs, which present a high correlation with

the lithofacies we are aiming to predict. These high–order features (from both borehole image logs) are extracted, concatenated and used in a k–nearest neighbour classifier to obtain the definitive categorical prediction of lithofacies class for each data sample.

With this architecture we achieved a global classification accuracy of over 97%. We tested our method predicting continous blocks of 100 samples each instead of single data points to obtain more robust and reliable results. Even though the number of samples for each class predicted was not uniform (ARGILITO and CALCILUTITO had far less samples than the rest of the classes) our classifier showed consistent results and no class was predicted with an accuracy inferior to 94%.

Taking into account the results we presented on this contribution and the current state of the art of automatic lithofacies prediction we consider that this novel method might be considered a promissing approach for producing reliable classifications when using borehole image data logs. Apart from that, it is also worth mentioning that this workflow does not require the user to have any previous advanced knowledge regarding deep networks nor artificial intelligence nor geology in order to use it.

It is important to notice that Deep Learning techniques and, specially, convolutional neural networks are expected to keep evolving and increasing their performance in the following years. Even though the results we obtained in this work are very interesting, it should be clear that the applicability that other types of deep learning techniques have on the oil/gas industry has not been exploited yet. The use of modern convolutional diagram–type architectures such as residual blocks or inception modules could make the classification results even better. For instance, generative adversarial networks could also be used to simulate borehole data which could be used to create more accurate petrophysical models; while deep intuition methods, such as the one shown by AlphaZero (Marcus, 2018; Barratt and Pan) (which after only 24 hours of self–taughthing itself to play the ancient chinese game GO defeated the world human champion), could help us automatize the procedures carried on by geologists and analysts regarding reservoir characterization, structure and formations detection or even assist in highly–qualified tasks such as deciding where to drill a borehole in the reservoir or whether it is worthy or feasible to extract oil from a reservoir or not.

7.5 Method for estimating petrophysical logs using stacked autoencoders and borehole image data logs

In Section 6, we introduced a novel method for estimating porosity and permeability logs from borehole image data, by using deep learning techniques. Our method is based on two identical 3–level stacked AE, one for the ultrasonic borehole data and another for the microresistivity borehole data, which were used to characterize the inner features of

these data logs. Later, these features are used in a SVM classifier to obtain an estimation of porosity or permeability values.

Previous work on this field had shown very positive results while using other types of wirelogs (not images) such as Gamma–Ray, Neutron Porosity and/or Bulk Density Logs. On the other hand, more recent research has used linear operations to predict permeability and porosity from borehole images by estimating fracture aperture (Cheung et al., 1990; Luthi and Souhaité, 1990) or by calculating the rock matrix porosity of ultrasonic borehole images (de Jesus et al., 2016). However the results they obtained achieved lower accuracy than the ones we obtained using the method we presented in this work.

Our method achieved an estimation correlation of over $R^2 = 96\%$ for both porosity and permeability logs estimations, when using the testing data presented in the previously mentioned section. This value presented a notable improvement regarding the benchmark, except for the result obtained by Mohaghegh et al. (1995), which had a slightly better result than ours. Even though it is important to notice that Mohaghegh et al. (1995): used a far smaller testing sample group (23 data samples) than ours (3700 sample points); and used other types of wirelogs which seem to be more intrisically related to the borehole lithology than borehole image logs (such as Gamma–Ray or Bulk Density Logs).

Apart from that, it is also interesting to mention that our work was based on brazilian pre–salt carbonate reservoir data. The data obtained from this group of basins has proven to be difficult to analyze as most of the well–known models currently used to analyze other types of reservoirs from all over the world do not seem to apply correctly to them. The brazilian pre–salt Ccarbonate is still considered a challenge for the petrophysical world for these and other reasons.

The increase in regression performance with our method is important not only because it allows petrophysicists to estimate the values of permeability and porosity logs more accurately, but also because it shows that borehole image data Llgs can really be linked to these petrophysical properties in such a way that the information extracted from them using deep methods can be used to estimate these logs with higher accuracy than the methods that use conventional wirelogs such as Gamma Ray, Spontaneous Potential or Bulk Density logs.

It is also important to emphasize that our method relies on the extraction of geological information for each type of borehole data within a certain sliding window, which preserves the geological and lithological structures that we expect to be related to the petrophysical logs we are estimating. To the best of our knowledge, all previous benchmark used single point operations to perform their estimations, rather than taking into account the information contained in the surroundings of each data sample.

Future work in this area includes testing this method using other deep learning architectures (e.g., convolutional, rectifier, normalization and pooling layers), and how the effect of convolutional methods affects the regression result, as well as studying how the

size of the sliding window affects the final prediction.

Bibliography

Bernt Sigve Aadnoy. Inversion technique to determine the in-situ stress field from fracturing data. *Journal of Petroleum Science and Engineering*, 4(2):127–141, 1990.

Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

Mohammad Abdideh. Estimation of permeability using artificial neural networks and regression analysis in an iran oil field. *International Journal of the Physical Sciences*, 7 (34):5308–5313, 2012.

Forest Agostinelli, Michael R Anderson, and Honglak Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

Mohammed Al-Khabbaz, Hanan Abu-Hebail, Mahmood Akbar, Badruzzaman Khan, Salem Al-Sabea, Mona Rashaid, Robert Laronga, Alexander Aviantara, Sandeep Chakravorty, and Rohoullah Dashti. Complex artifacts in resistivity and acoustic image data: Recognition and use in borehole image interpretation. In *AAPG Annual Convention and Exhibition*, Long Beach, California, Apr. 22-25 2012. URL http://www.searchanddiscovery.com/documents/2012/40975alkhabbaz/ndx_alkhabbaz.pdf.

Jaouher Ben Ali, Nader Fnaiech, Lotfi Saidi, Brigitte Chebel-Morello, and Farhat Fnaiech. Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, 89:16–27, 2015.

Jean Allard and Noureddine Atalla. *Propagation of sound in porous media: modelling sound absorbing materials 2e*. John Wiley & Sons, 2009.

Jont B Allen. How do humans process and recognize speech? *IEEE Transactions on speech and audio processing*, 2(4):567–577, 1994.

Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

and others. Nuclear magnetism well logging, June 2 1964. US Patent 3,135,912.

Gene Anderson. Coring and core analysis handbook. 1975.

James A Anderson and Edward Rosenfeld. *Talking nets: An oral history of neural networks*. MiT Press, 2000.

GP Angeleri and R Carpi. Porosity prediction from seismic data. *Geophysical prospecting*, 30(5):580–607, 1982.

Fatai Anifowose and Abdulazeez Abdulraheem. Fuzzy logic-driven and svm-driven hybrid computational intelligence models applied to oil and gas reservoir characterization. *Journal of Natural Gas Science and Engineering*, 3(3):505–517, 2011.

Andreas Antoniou. *Digital signal processing*. McGraw-Hill, 2016.

RP API. 40, recommended practices for core analysis, second, 2010.

Gustave E Archie et al. The electrical resistivity log as an aid in determining some reservoir characteristics. *Transactions of the AIME*, 146(01):54–62, 1942.

Said Assous, Peter Elkington, Stuart Clark, and James Whetton. Automated detection of planar geologic features in borehole images. *Geophysics*, 79(1):D11–D19, 2013.

Peter Auer, Harald Burgsteiner, and Wolfgang Maass. A learning rule for very simple universal approximators consisting of a single layer of perceptrons. *Neural networks*, 21 (5):786–795, 2008.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Pierre Baldi. Gradient descent learning algorithm overview: A general dynamical systems perspective. *IEEE Transactions on neural networks*, 6(1):182–195, 1995.

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49, 2012.

Jeffrey Barratt and Chuanbo Pan. Deep imitation learning for playing real time strategy games.

Colleen A Barton, Mark D Zoback, and Kerry L Burns. In-situ stress orientation and magnitude at the fenton geothermal site, new mexico, determined from wellbore breakouts. *Geophysical Research Letters*, 15(5):467–470, 1988.

Colleen A Barton, Daniel Moos, Pavel Peska, Mark D Zoback, et al. Utilizing wellbore image data to determine the complete stress tensor: application to permeability anisotropy and wellbore stability. *The log analyst*, 38(06), 1997.

Tanwi Basu, Robert Dennis, BD Al-Khobar, W Al Awadi, SJ Isby, E Vervest, and R Mukherjee. Automated facies estimation from integration of core, petrophysical logs, and borehole images. In *AAPG Annual Meeting*, pages 1–7, 2002.

Markus Bath. *Introduction to seismology*, volume 27. Birkhäuser, 2013.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

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

Randy Beste, T Hagiwara, George King, Robert Strickland, GA Merchant, et al. A new high resolution array induction tool. In *SPWLA 41st Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 2000.

Brain Leke Betechuoh, Tshilidzi Marwala, and Thando Tettey. Autoencoder networks for hiv classification. *Current Science*, pages 1467–1473, 2006.

Arpit Bhardwaj and Aruna Tiwari. Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, 42(10):4611–4620, 2015.

Ed L Bigelow. *Introduction to Wireline Log Analysis*. Baker Atlas, 2002.

G. R. Blake. Bulk density. *Methods of soil analysis. Part 1. Physical and mineralogical properties, including statistics of measurement and sampling*, (methodsofsoilana):374–390, 1965.

Miguel Bosch, Tapan Mukerji, and Ezequiel F Gonzalez. Seismic inversion for reservoir properties combining statistical rock physics and geostatistics: A review. *Geophysics*, 2010.

L Bourke, P Delfiner, JC Trouiller, T Fett, M Grace, S Luthi, O Serra, and E Standen. Using formation microscanner images. *The Technical Review*, 37(1):16–40, 1989.

Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw Hill New York, 1986.

DR Brouwer, Geir Nævdal, JD Jansen, Erland H Vefring, CPJW Van Kruijsdijk, et al. Improved reservoir management through optimal control and continuous model updating. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2004.

Matthew Brown and David G Lowe. Invariant features from interest point groups. In *BMVC*, volume 4, 2002.

NeT Burdine et al. Relative permeability calculations from pore size distribution data. *Journal of Petroleum Technology*, 5(03):71–78, 1953.

M Burshears, TJ O'brien, RD Malone, et al. A multi-phase, multi-dimensional, variable composition simulation of gas production from a conventional gas reservoir in contact with hydrates. In *SPE Unconventional Gas Technology Symposium*. Society of Petroleum Engineers, 1986.

Leonid Buryakovsky, George V Chilingar, Herman H Rieke, and Sanghee Shin. *Fundamentals of the petrophysics of oil and gas reservoirs*. John Wiley & Sons, 2012.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

Hua Chai, Ning Li, Chengwen Xiao, Xingli Liu, Duoli Li, Caizhi Wang, and Dacheng Wu. Automatic discrimination of sedimentary facies and lithologies in reef-bank reservoirs using borehole image logs. *Applied Geophysics*, 6(1):17–29, 2009.

Roland Chemali, Stan Gianzero, SM Su, et al. The dual laterolog in common complex situations. In *SPWLA 29th Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 1988.

CH Cheng, FL Paillet, WD Pennington, et al. Acoustic-waveform logging-advances intheory and application. *The Log Analyst*, 33(03), 1992.

PS-Y Cheung, D Heliot, et al. Workstation-based fracture evaluation using borehole images and wireline logs. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 1990.

Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2392–2396. IEEE, 2017.

Satinder Chopra and Kurt J Marfurt. *Seismic attributes for prospect identification and reservoir characterization*. Society of Exploration Geophysicists and European Association of Geoscientists and Engineers, 2007.

Charles K Chui. Wavelets: a tutorial in theory and applications. *Wavelet Analysis and its Applications, San Diego, CA: Academic Press,/ c1992, edited by Chui, Charles K.*, 1992.

Charles K Chui. *An introduction to wavelets*. Elsevier, 2016.

H Kerem Cigizoglu. Estimation and forecasting of daily suspended sediment data by multi-layer perceptrons. *Advances in Water Resources*, 27(2):185–195, 2004.

David A Clausi and M Ed Jernigan. Designing gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849, 2000.

A Cochocki and Rolf Unbehauen. *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc., 1993.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

Milton Craft and Dare Keelan. Coring. part 7–analytical aspects of sidewall coring. *World Oil (Sept. 1985)*, pages 77–90.

Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

Kevin J Cunningham, Janine I Carlson, and Neil F Hurley. New method for quantification of vuggy porosity from digital optical borehole images as applied to the karstic pleistocene limestone of the biscayne aquifer, southeastern florida. *Journal of Applied Geophysics*, 55(1):77–90, 2004.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

Carlos Tadeu da Costa Fraga, Capeleiro Pinto, Antonio Carlos, Celso Cesar Moreira Branco, Cezar Augusto da Silva Paulo, et al. Brazilian pre-salt: an impressive journey from plans and challenges to concrete results. In *Offshore Technology Conference*. Offshore Technology Conference, 2015.

Amirmasoud Kalantari Dahaghi et al. Numerical simulation and modeling of enhanced gas recovery and co2 sequestration in shale gas reservoirs: A feasibility study. In *SPE international conference on CO2 capture, storage, and utilization*. Society of Petroleum Engineers, 2010.

Padideh Danaee, Reza Ghaeini, and David A Hendrix. A deep learning approach for cancer detection and relevant gene identification. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2017*, pages 219–229. World Scientific, 2017.

Henry Philibert Gaspard Darcy. *Les Fontaines publiques de la ville de Dijon. Exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau, etc.* 1856.

Ingrid Daubechies et al. Ten lectures on wavelets. In *CBMS-NSF regional conference series in applied mathematics*, volume 61, 1991.

Candida Menezes de Jesus, Andre Luiz Martins Compan, Rodrigo Surmas, et al. Permeability estimation using ultrasonic borehole image logs in dual-porosity carbonate reservoirs. *Petrophysics*, 57(06):620–637, 2016.

Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

Philippe Doyen. *Seismic reservoir characterization: An earth modelling perspective*, volume 2. EAGE publications Houten, 2007.

P-C Du, GA Mansoori, et al. A continuous mixture computational algorithm for reservoir fluids phase behavior. In *SPE California Regional Meeting*. Society of Petroleum Engineers, 1986.

K-J Dunn, David J Bergman, and Gerald A LaTorraca. *Nuclear magnetic resonance: Petrophysical and logging applications*, volume 32. Elsevier, 2002.

LOUISE S. DURHAM. An unconventional idea open to interpretation, 2013. http://archives.aapg.org/explorer/2013/03mar/seismic_int0313.cfm.

Sean R Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.

Michael Egmont-Petersen, Dick de Ridder, and Heinz Handels. Image processing with neural networks—a review. *Pattern recognition*, 35(10):2279–2301, 2002.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940, 2016.

Salaheldin Elkatatny, Mohamed Mahmoud, Zeeshan Tariq, and Abdulazeez Abdulraheem. New insights into the prediction of heterogeneous carbonate reservoir permeability from well logs using artificial intelligence network. *Neural Computing and Applications*, pages 1–11, 2017.

Darwin V Ellis. Neutron porosity devices-what do they measure? *First Break*, 4(3):11–17, 1986.

Darwin V Ellis and Julian M Singer. *Well logging for earth scientists*, volume 692. Springer, 2007.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

Mohammad Hemmat Esfe, Masoud Afrand, Somchai Wongwises, Ali Naderi, Amin Asadi, Sara Rostami, and Mohammad Akbari. Applications of feedforward multilayer perceptron artificial neural networks and empirical correlation for prediction of thermal conductivity of mg (oh) 2–eg using experimental data. *International Communications in Heat and Mass Transfer*, 67:46–50, 2015.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

Leonhard Euler et al. *Introductio in analysin infinitorum...; tomus primus*. Lausannae: apud Marcum-Michaelem Bousquet & Socies, 1748.

Xue Feng, Yaodong Zhang, and James Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1759–1763. IEEE, 2014.

LL Gadekea, ML Gartner, DE Sharbak, DF Wyatt, et al. The interpretation of radioactive-tracer logs using gamma-ray spectroscopy measurements. *The Log Analyst*, 32(01), 1991.

Philippe Gaillot, Tim Brewer, Philippe Pezard, and En-Chao Yeh. Borehole imaging tools- principles and applications. *Scientific Drilling*, 5:1–4, 2007.

Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14):2627–2636, 1998.

J Garson, A Aggarwal, and S Sarkar. Resnet ver 1.2 manual. *Biodiversity and Biocultural Conservation Laboratory, University of Texas, Austin*, 2002.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

Rafael Gonzalez, Richard Woods, and Steven Eddins. Digital image processing using matlab gatesmark publishing. 2009.

Rafael C Gonzalez, Richard E Woods, et al. Digital image processing, 1992.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Travis R Goodwin and Sanda M Harabagiu. Deep learning from eeg reports for inferring underspecified information. *AMIA Summits on Translational Science Proceedings*, 2017: 112, 2017.

Steven Greenberg, William A Ainsworth, Arthur N Popper, and Richard R Fay. *Speech processing in the auditory system*, volume 18. Springer, 2004.

Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. Comparison of texture features based on gabor filters. *IEEE Transactions on Image processing*, 11(10):1160–1167, 2002.

Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.

HC Haimson, CG Herrick, et al. Borehole breakouts-a new tool for estimating in situ stress? In *ISRM International Symposium*. International Society for Rock Mechanics, 1986.

Jonathan Hall, Marco Ponzi, Mauro Gonfalini, Giorgio Maletti, et al. Automatic extraction and characterisation of geological features and textures from borehole images and core photographs. In *SPWLA 37th Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 1996.

GM Hamada and MA Elshafei. Neural network prediction of porosity and permeability of heterogeneous gas sand reservoirs using nmr and conventional logs. *Nafta*, 61(10): 451–460, 2010.

Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

Bob A Hardage. Vertical seismic profiling. *The Leading Edge*, 4(11):59–59, 1985.

Mohamad H Hassoun. *Fundamentals of artificial neural networks*. MIT press, 1995.

Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

AJ Hayman, Philippe Parent, Philip Cheung, Patrick Verges, et al. Improved borehole imaging by ultrasonics. *SPE production & facilities*, 13(01):5–14, 1998.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Donald Olding Hebb. *The organization of behavior: A neuropsychological approach*. John Wiley & Sons, 1949.

MM Herron and SL Herron. Geological applications of geochemical well logging. *Geological Society, London, Special Publications*, 48(1):165–175, 1990.

R Hertzog, L Colson, O Seeman, M O'Brien, H Scott, D McKeon, P Wraight, J Grau, D Ellis, J Schweitzer, et al. Geochemical logging with spectrometry tools. *SPE Formation Evaluation*, 4(02):153–162, 1989.

Yashar D Hezaveh, Laurence Perreault Levasseur, and Philip J Marshall. Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature*, 548 (7669):555, 2017.

Gregory Hickok and David Poeppel. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8(5):393, 2007.

Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

Mehdi Matt Honarpour, F Koederitz, and A Herbert. Relative permeability of petroleum reservoirs. 1986.

Jeffrey R Hook et al. An introduction to porosity. *Petrophysics*, 44(03), 2003.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Guang-Bin Huang, Yan-Qiu Chen, and Haroon A Babri. Classification ability of single hidden layer feedforward neural networks. *IEEE Transactions on Neural Networks*, 11 (3):799–801, 2000.

David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

Baker Hughes. Introduction to wireline log analysis. *Baker Atlas*, 2002.

Jordi Inglada. Automatic recognition of man-made objects in high resolution optical remote sensing images by svm classification of geometric image features. *ISPRS journal of photogrammetry and remote sensing*, 62(3):236–248, 2007.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

EE Java. Large scale visual recognition challenge 2016 (ilsvrc2016).

Meijuan Jiang and Kyle T Spikes. Rock-physics and seismic-inversion based reservoir characterization of the haynesville shale. *Journal of Geophysics and Engineering*, 13(3): 220, 2016.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

Abraham K John, Larry Wayne Lake, Carlos Torres-Verdin, Sanjay Srinivasan, et al. Seismic-based facies identification and classification using simple statistics. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2005.

Vijay John, Seiichi Mita, Zheng Liu, and Bin Qi. Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks. In *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, pages 246–249. IEEE, 2015.

Cleveland M Jones, Hernani Chaves, et al. Assessment of yet-to-find oil in the brazilian pre-salt region. In *Brasil Offshore*. Society of Petroleum Engineers, 2011.

Donald G Jorgensen et al. Estimating permeability in water-saturated formations. *The Log Analyst*, 29(06), 1988.

Matthias Jungmann, Margarete Kopal, Christoph Clauser, and Thomas Berlage. Multi-class supervised classification of electrical borehole wall images using texture features. *Computers & geosciences*, 37(4):541–553, 2011.

Meina Kan, Shiguang Shan, Hong Chang, and Xilin Chen. Stacked progressive auto-encoders (spae) for face recognition across poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1883–1890, 2014.

Dare K Keelan et al. Core analysis techniques and applications. In *SPE Eastern Regional Meeting*. Society of Petroleum Engineers, 1972.

Mohan Kelkar, Godofredo Perez, and Anil Chopra. *Applied geostatistics for reservoir characterization*. Society of petroleum engineers Richardson, TX, 2002.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Akisato Kimura, Ichiro Takahashi, Masaomi Tanaka, Naoki Yasuda, Naonori Ueda, and Naoki Yoshida. Single-epoch supernova classification with deep convolutional neural networks. In *Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on*, pages 354–359. IEEE, 2017.

Robert L Kleinberg and Jasper A Jackson. An introduction to the history of nmr well logging. *Concepts in Magnetic Resonance*, 13(6):340–342, 2001.

Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Handwritten digit recognition by neural networks with single-layer training. *IEEE Transactions on neural networks*, 3 (6):962–968, 1992.

A Koponen, M Kataja, and J Timonen. Permeability and effective porosity of porous media. *Physical Review E*, 56(3):3319, 1997.

Josef Kozeny. *Über kapillare leitung des wassers im boden:(aufstieg, versickerung und anwendung auf die bewässerung)*. Hölder-Pichler-Tempsky, 1927.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Ben Kröse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks. 1993.

BR Kulander and SL Dean. Drilling-induced fractures. Technical report, American Association of Petroleum Geologists, Tulsa, OK, 1988.

Rajesh Kumar, RK Aggarwal, and JD Sharma. Comparison of regression and artificial neural network models for estimation of global solar radiations. *Renewable and Sustainable Energy Reviews*, 52:1294–1299, 2015.

H Laastad, E Haukefaer, S Young, A Tehrani, P Delaunay, K Feather, J Phillips, S Marca, et al. Water-based formation imaging and resistivity logging in oil-based drilling fluids-today's reality. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2000.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2169–2178. IEEE, 2006.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

MoC Leverett et al. Capillary behavior in porous solids. *Transactions of the AIME*, 142 (01):152–169, 1941.

Hao Li and Siddharth Misra. Prediction of subsurface nmr t2 distributions in a shale petroleum system using variational autoencoder-based neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2395–2397, 2017.

Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172: 1–11, 2018.

Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection1. *Computers & security*, 21(5):439–448, 2002.

Schlumberger Limited. *Schlumberger log interpretation charts*. Schlumberger, 1984.

P Lin, XL Li, YM Chen, and Y He. A deep convolutional neural network architecture for boosting image discrimination accuracy of rice species. *Food and Bioprocess Technology*, pages 1–9, 2018.

Margarete Linek, Matthias Jungmann, Thomas Berlage, Renate Pechnig, and Christoph Clauser. Rock classification based on resistivity patterns in electrical borehole wall images. *Journal of Geophysics and Engineering*, 4(2):171, 2007.

Larry R Lines, Alton K Schultz, and Sven Treitel. Cooperative inversion of geophysical data. *Geophysics*, 53(1):8–20, 1988.

Richard P Lippmann. Speech recognition by machines and humans. *Speech communication*, 22(1):1–15, 1997.

Paulo J Lisboa and Azzam FG Taktak. The use of artificial neural networks in decision support in cancer: a systematic review. *Neural networks*, 19(4):408–415, 2006.

GA Lock, WA Hoyer, et al. Carbon-oxygen (c/o) log: Use and interpretation. *Journal of Petroleum Technology*, 26(09):1–044, 1974.

JC Lofts and LT Bourke. The recognition of artefacts from acoustic and resistivity borehole imaging devices. *Geological Society, London, Special Publications*, 159(1):59–76, 1999.

David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

Jiwen Lu, Gang Wang, and Jie Zhou. Simultaneous feature and dictionary learning for image set based face recognition. *IEEE Transactions on Image Processing*, 2017.

F Jerry Lucia. *Carbonate reservoir characterization: an integrated approach*. Springer Science & Business Media, 2007.

Stefan M Luthi and Philippe Souhaité. Fracture apertures from electrical borehole scans. *Geophysics*, 55(7):821–833, 1990.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.

Gian Luca Marcialis and Fabio Roli. Fusion of multiple fingerprint matchers by single-layer perceptron with class-separation loss function. *Pattern Recognition Letters*, 26 (12):1830–1839, 2005.

Gary Marcus. Innateness, alphazero, and artificial intelligence. *arXiv preprint arXiv:1801.05667*, 2018.

Benjamin Marlin, Kevin Swersky, Bo Chen, and Nando Freitas. Inductive principles for restricted boltzmann machine learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 509–516, 2010.

Sarfaraz Masood, Shubham Gupta, Abdul Wajid, Suhani Gupta, and Musheer Ahmed. Prediction of human ethnicity from facial images using neural networks. In *Data Engineering and Intelligent Computing*, pages 217–226. Springer, 2018.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Colin McPhee, Jules Reed, and Izaskun Zubizarreta. *Core Analysis: A Best Practice Guide*, volume 64. Elsevier, 2015.

Zhong Meng, Shinji Watanabe, John R Hershey, and Hakan Erdogan. Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 271–275. IEEE, 2017.

Stefan Menger et al. New aspects of the borehole televiewer decentralization correction. *The log analyst*, 35(04), 1994.

R Benton Metcalf, M Meneghetti, Camille Avestruz, Fabio Bellagamba, Clécio R Bom, Emmanuel Bertin, Rémi Cabanac, Etienne Decencière, Rémi Flamary, Raphael Gavazzi, Manuel B Valentín, et al. The strong gravitational lens finding challenge. *arXiv preprint arXiv:1802.03609*, 2018.

Yves Meyer. Wavelets-algorithms and applications. *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p.*, 1993.

WR Mills, David C Stromswold, LS Allen, et al. Pulsed neutron porosity logging. In *SPWLA 29th Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 1988.

Marvin Lee Minsky. *Theory of neural-analog reinforcement systems and its application to the brain model problem*. Princeton University., 1954.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, page bbx044, 2017.

Shahab Mohaghegh, Reza Arefi, Samuel Ameri, MH Hefner, et al. A methodological approach for reservoir heterogeneity characterization using artificial neural networks. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 1994.

Shahab Mohaghegh, Reza Arefi, Samuel Ameri, D Rose, et al. Design and development of an artificial neural network for estimation of formation permeability. *SPE Computer Applications*, 7(06):151–154, 1995.

Shahab Mohaghegh, Reza Arefi, Sam Ameri, Khashayar Aminiand, and Roy Nutter. Petroleum reservoir characterization with the aid of artificial neural networks. *Journal of Petroleum Science and Engineering*, 16(4):263–274, 1996.

Shahab Mohaghegh, Bogdan Balan, Samuel Ameri, et al. Permeability determination from well log data. *SPE formation evaluation*, 12(03):170–174, 1997.

Abdel-rahman Mohamed, Tara N Sainath, George Dahl, Bhuvana Ramabhadran, Geoffrey E Hinton, and Michael A Picheny. Deep belief networks using discriminative features for phone recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5060–5063. IEEE, 2011.

Stephen Morris, Jeremy C Lofts, Gavin Lindsay, Christian Fulda, John Dahl, et al. High-quality electrical borehole images while-drilling: Increased confidence in well positioning and drilling-hazard mitigation from real-time data. In *IADC/SPE Drilling Conference*. Society of Petroleum Engineers, 2006.

Kohei Mukaihara, Sakriani Sakti, and Satoshi Nakamura. Recognizing emotionally coloured dialogue speech using speaker-adapted dnn-cnn bottleneck features. In *International Conference on Speech and Computer*, pages 632–641. Springer, 2017.

T Mukerji, A Jørstad, P Avseth, G Mavko, and JR Granli. Mapping lithofacies and pore-fluid probabilities in a north sea reservoir: Seismic inversions and statistical rock physics. *Geophysics*, 66(4):988–1001, 2001.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

Jamal M Nazzal, Ibrahim M El-Emary, and Salam A Najim. Multilayer perceptron neural network (mlps) for analyzing the properties of jordan oil shale 1. 2008.

Bill M Newberry, Steven M Hansen, and Tom T Perrett. A method for analyzing textural changes within clastic environments utilizing electrical borehole images. 2004.

Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com/.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

CM Oldenburg, Karsten Pruess, and Sally M Benson. Process modeling of co2 injection into natural gas reservoirs for carbon sequestration and enhanced gas recovery. *Energy & Fuels*, 15(2):293–298, 2001.

Dean S Oliver, Albert C Reynolds, and Ning Liu. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, 2008.

El Ouahed, A Kouider, Djebbar Tiab, Amine Mazouzi, Sarfraz A Jokhio, et al. Application of artificial intelligence to characterize naturally fractured reservoirs. In *SPE International Improved Oil Recovery Conference in Asia Pacific*. Society of Petroleum Engineers, 2003.

Ahmed Ouenes. Practical application of fuzzy logic and neural networks to fractured reservoir characterization. *Computers & Geosciences*, 26(8):953–962, 2000.

Sungheon Park and Nojun Kwak. Analysis on the dropout effect in convolutional neural networks. In *Asian Conference on Computer Vision*, pages 189–204. Springer, 2016.

Sybil P Parker. *McGraw-Hill concise encyclopedia of science & technology*. McGraw-Hill, 1984.

Dan W Patterson. *Artificial neural networks: theory and applications*. Prentice Hall PTR, 1998.

Daniel Peralta, Isaac Triguero, Salvador García, Yvan Saeys, Jose M Benitez, and Francisco Herrera. On the use of convolutional neural networks for robust classification

of multiple fingerprint captures. *International Journal of Intelligent Systems*, 33(1): 213–230, 2018.

Pierre Peretto. *An introduction to the modeling of neural networks*, volume 2. Cambridge University Press, 1992.

Hector H Perez, Akhil Datta-Gupta, Srikanta Mishra, et al. The role of electrofacies, lithofacies, and hydraulic flow units in permeability predictions from well logs: a comparative analysis using classification trees. *SPE Reservoir Evaluation & Engineering*, 8 (02):143–155, 2005.

Ritesh Pradhan, Ramazan S Aygun, Manil Maskey, Rahul Ramachandran, and Daniel J Cecil. Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Transactions on Image Processing*, 27(2):692–702, 2018.

Stephen E Prensky. Advances in borehole imaging technology and applications. *Geological Society, London, Special Publications*, 159(1):1–43, 1999.

Michael J Pyrcz and Clayton V Deutsch. *Geostatistical reservoir modeling*. Oxford university press, 2014.

II Rabi, JR Zacharias, S Millman, and P Kusch. Milestones in magnetic resonance:'a new method of measuring nuclear magnetic moment'. 1938., 1991.

Baouche Rafik and Baddari Kamel. 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*, 2016.

Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.

Trygve Randen and John Hakon Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on pattern analysis and machine intelligence*, 21(4):291–310, 1999.

Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Josef P Rauschecker and Sophie K Scott. Maps and streams in the auditory cortex: nonhuman primates illuminate human speech processing. *Nature neuroscience*, 12(6): 718, 2009.

Rishi R Rawat, Daniel Ruderman, David B Agus, and Paul Macklin. Deep learning to determine breast cancer estrogen receptor status from nuclear morphometric features in h&e images, 2017.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

S Duffy Russell, Mahmoud Akbar, Badarinadh Vissapragada, and Gordon M Walkden. Rock types and permeability prediction from dipmeter and image logs: Shuaiba reservoir (aptian), abu dhabi. *AAPG bulletin*, 86(10), 2002.

Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. *Artificial Neural Networks–ICANN 2010*, pages 92–101, 2010.

Schlumberger. Oilfield review, 2011. http://www.slb.com/~/media/Files/resources/oilfield_review/ors11/spr11/composite.pdf.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Hongbin Shen and Kuo-Chen Chou. Using optimized evidence-theoretic k-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. *Biochemical and biophysical research communications*, 334(1):288–292, 2005.

Hoo-Chang Shin, Matthew R Orton, David J Collins, Simon J Doran, and Martin O Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1930–1943, 2013.

AM Sibbit, O Faivre, et al. The dual laterolog response in fractured rocks. In *SPWLA 26th Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 1985.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Roger M Slatt. *Stratigraphic reservoir characterization for petroleum geologists, geophysicists, and engineers*, volume 61. Elsevier, 2006.

Steven W Smith et al. The scientist and engineer's guide to digital signal processing. 1997.

A Sophian, GY Tian, D Taylor, and J Rudlin. Electromagnetic and eddy current ndt: a review. *Insight*, 43(5):302–306, 2001.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Schlumberger Staff. Fundamentals of formation testing, 2006.

Elias M Stein and Guido Weiss. *Introduction to Fourier analysis on Euclidean spaces (PMS-32)*, volume 32. Princeton university press, 2016.

Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

HL Stone. Estimation of three-phase relative permeability and residual oil data. *J. Pet. Technol.;(United States)*, 12(4), 1973.

Bernard Storr. Porosimeter and methods of assessing porosity, January 12 1988. US Patent 4,718,270.

Roger H Stuewer. The compton effect. turning points in physics. *The Compton effect. Turning points in physics., by Stuewer, RH. New York, NY (USA): Science History Publications (Neale Watson), 12+ 368 p.*, 1975.

J Sugiura. Improved lwd density images in the presence of cyclic borehole noise. AADE-11-NTCE-66 presented at the 2011 National Technical Conference and Exhibition, Houston, Texas, USA, 12-14 April, 2011.

Zehang Sun, George Bebis, and Ronald Miller. On-road vehicle detection using gabor filters and support vector machines. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, volume 2, pages 1019–1022. IEEE, 2002.

Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, 21(9):1120–1124, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Chun Chet Tan and Chikkannan Eswaran. Using autoencoders for mammogram compression. *Journal of medical systems*, 35(1):49–58, 2011.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

EC Thomas. 50th anniversary of the archie equation: Archie left more than just an equation. *The Log Analyst (May–June)*, 199, 1992.

Aytekin Timur et al. An investigation of permeability, porosity, and residual water saturation relationships. In *SPWLA 9th annual logging symposium*. Society of Petrophysicists and Well-Log Analysts, 1968.

M Tingay, J Reinecker, and B Müller. Borehole breakout and drilling-induced fracture analysis from image logs. *World Stress Map Project*, pages 1–8, 2008.

J Tittman, H Sherman, WA Nagel, et al. The sidewall epithermal neutron porosity log. *Journal of Petroleum Technology*, 18(10):1–351, 1966.

MP Tixier. Evaluation of permeability from electric-log resistivity gradients. *Oil and Gas Journal*, 48(6):113–123, 1949.

Yoshihiro Todoroki, Xian-Hua Han, Yutaro Iwamoto, Lanfen Lin, Hongjie Hu, and Yen-Wei Chen. Detection of liver tumor candidates from ct images using deep convolutional neural networks. In *International Conference on Innovation in Medicine and Healthcare*, pages 140–145. Springer, 2017.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008.

Michael Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on image processing*, 4(11):1549–1560, 1995.

M Blanco Valentin. Ultrasonic pad removal method: In progress. 2018.

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

Paolo Vecchiotti, Fabio Vesperini, Emanuele Principi, Stefano Squartini, and Francesco Piazza. Convolutional neural networks with 3-d kernels for voice activity detection in a multiroom environment. In *Multidisciplinary Approaches to Neural Computing*, pages 161–170. Springer, 2018.

Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

Fjodor Van Veen. The neural network zoo, 2016. http://www.asimovinstitute.org/neural-network-zoo/.

Bolivia Vega, Abhishek Dutta, and Anthony R Kovscek. Ct imaging of low-permeability, dual-porosity systems using high x-ray contrast gas. *Transport in porous media*, 101(1): 81–97, 2014.

Weixing Wang. An edge based segmentation algorithm for rock fracture tracing. In *Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on*, pages 43–48. IEEE, 2005.

Graham A Webb. *Annual reports on NMR spectroscopy*, volume 80. Academic Press, 2013.

Hilary Weller. Fourier analysis. 2015.

Stefan Wessling and Jianyong Pei. System and method for automatic detection and analysis of borehole breakouts from images and the automatic generation of alerts, February 24 2015. US Patent 8,965,701.

Stefan Wessling, Thomas Dahl, Dinah Margaret Pantic, et al. Challenges and solutions for automated wellbore status monitoring-breakout detection as an example. In *SPE Digital Energy Conference and Exhibition*. Society of Petroleum Engineers, 2011.

JH Williams and CD Johnson. Borehole-wall imaging with acoustic and optical televiewers for fractured-bedrock aquifer investigations. In *Proceedings of the 7th Minerals and Geotechnical Logging Symposium*, volume 53. Golden:[sn], 2000.

Gail Williamson, MA Lovell, and Peter K Harvey. Borehole imaging: Applications and case histories. Geological Society of London, 1999.

Cort J Willmott. On the validation of models. *Physical geography*, 2(2):184–194, 1981.

Zhirong Wu, Yinda Zhang, Fisher Yu, and Jianxiong Xiao. A gpu implementation of googlenet, 2014.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Jun Xu, Lei Xiang, Qingshan Liu, Hannah Gilmore, Jianzhong Wu, Jinghai Tang, and Anant Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE transactions on medical imaging*, 35(1):119–130, 2016.

Hongmei Yan, Yingtao Jiang, Jun Zheng, Chenglin Peng, and Qinghui Li. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, 30(2):272–281, 2006.

Kewen Yan, Shaohui Huang, Yaoxian Song, Wei Liu, and Neng Fan. Face recognition based on convolution neural network. In *Control Conference (CCC), 2017 36th Chinese*, pages 4077–4081. IEEE, 2017.

Shin-Ju Ye, Philippe Rabiller, Noomane Keskes, et al. Automatic high resolution texture analysis on borehole imagery. In *SPWLA 39th Annual Logging Symposium*. Society of Petrophysicists and Well-Log Analysts, 1998.

B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

Öz Yilmaz. *Seismic data analysis: Processing, inversion, and interpretation of seismic data*. Society of exploration geophysicists, 2001.

Dong Yu and Li Deng. Deep learning and its applications to signal and information processing [exploratory dsp]. *IEEE Signal Processing Magazine*, 28(1):145–154, 2011.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025. IEEE, 2011.

Xin-yun Zhang, Jing-nong Wang, Yan-jun Guo, et al. Advances and trends in logging while drilling technology. *Well Logging Technology*, 30(1):10, 2006.

Yu Zhang, William Chan, and Navdeep Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 4845–4849. IEEE, 2017.

Yuhang Zhang, Kun Fu, Hao Sun, Xian Sun, XinWei Zheng, and Hongqi Wang. A multi-model ensemble method based on convolutional neural networks for aircraft detection in large remote sensing images. *Remote Sensing Letters*, 9(1):11–20, 2018.

Shusen Zhou, Qingcai Chen, and Xiaolong Wang. Discriminative deep belief networks for image classification. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1561–1564. IEEE, 2010.

Karel Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics gems IV*, pages 474–485. Academic Press Professional, Inc., 1994.

Mark H Zweig and Gregory Campbell. Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical chemistry*, 39(4):561–577, 1993.

APPENDIX A

Common Examples of Convolutional Neural Networks Architectures

Each one of the types of layers introduced before can be seen as an important building block to be used to build CNNs. And even though knowing their specific functionalities helps in the process of CNN implementation, it still requires a lot of expertise and trial and error in order to fully get the sense of how these building blocks should be combined and used together to form an effective architecture which can be reliable for the specific problem to be solved. For that reason, it has been considered interesting to include in this work the most common CNN architectures that have appeared on the literature since Yann LeCun's first paper.

A.1 LeNet–5 (1998) – *Yann LeCun et al.*

In Figure A.1 the architecture of what is usually considered the first CNN ever is shown: LeCun's et al. *LeNet–5* Network. This net included a total of 7 layers: 2 convolutional layers, 2 pool layers, 2 FC layers and a final discriminative layer. LeNet–5 was used for classificating $[32 \times 32]$ hand–written digits images.

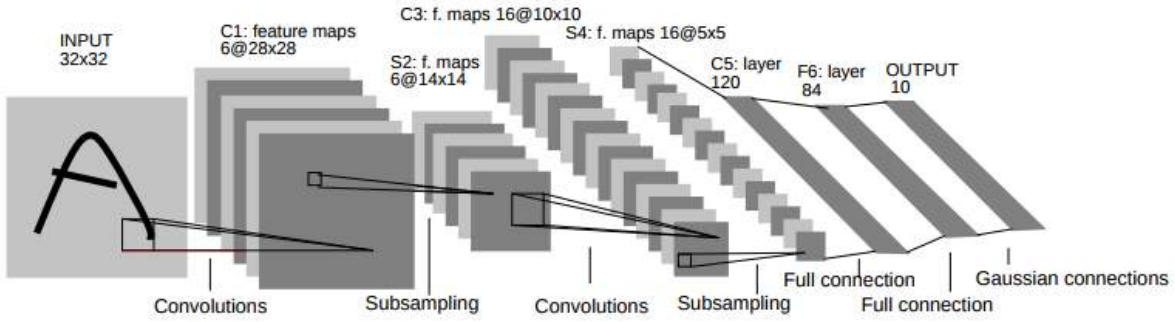


Figure A.1 – Architecture of LeNet–5, the CNN created by LeCun et al. in 1998, designed for hand–written digits classification and the first CNN ever implemented. Image extracted from LeCun et al. paper LeCun et al. (1998).

A.2 AlexNet (2012) – *Krizhevsky et al.*

Even though LeCun's net already used convolutional layers to extract features from input images, it was not until Krizhevsky et al. presented their net to the ILSVR Challenge in 2012 that CNNs started to be considered as real tools for image processing. Figure A.2 shows the architecture of 2012 AlexNet, which was mainly composed by 3 convolutional–pooling groups (ConvLayer + ReLU + Local Response Normalization Layer + Max–Pool Layer), 2 simple convolutional groups (ConvLayer + ReLU), 3 FC groups (FC + ReLU + Dropout) and a final discriminative layer that classified the images among different 1000 classes. This configuration won the 2012 ILSVR Challenge, with a total of 15.4% error on classification (that is 11% less error than the second best net).

Since LeCun et al., almost all CNN–like architecture that had been built and tested had used the ConvLayer + PoolLayer group. It was not until AlexNet that it was shown that it was possible (and equally effective) to stack some of these convolutional groups without using any pool or subsampling method after the convolution. On the other hand, Krizhevsky et al. implemented for the first time the dropout layer, as well as used data augmentation[1] to prevent overfitting.

[1]Data augmentation is an overfit–preventing technique which consists on using the same image samples repeatedly, changing some spatial parameters such as rotation, translation, reflections, etc. This way more data is input to the net and therefore the training process can be more complete, thus obtaining classification or recognition models that do not depend so strongly on these spatial features, but just on the shapes and real properties of the images we are aiming to characterize with the CNN.

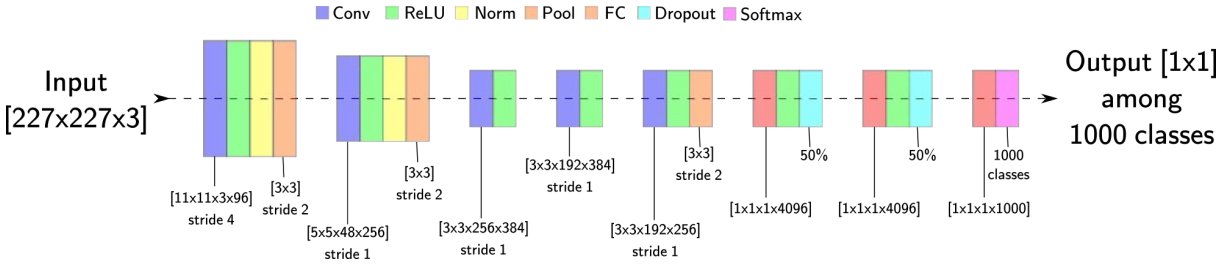


Figure A.2 – Illustration showing the architecture of Krizhevsky's (et al.) CNN: AlexNet. This CNN was the winner of 2012 ILSVR Challenge with an error rate of 15.4%. As it can be seen, it was formed by 8 main groups os layers: 5 convolutionals and 3 fully–connected.

A.3 ZF Net (2013) – *M. Zeiler and R. Fergus*

In 2013, the winner of the ILSVR Challenge was again a CNN model proposed by M. Zeiler and R. Fergus and called *ZF Net*. The architecture of ZFNet itself, which is shown in Figure A.3, was simply an improvement of the 2012 version of AlexNet. The main differences are that ZFNet used $[7 \times 7]$ filters with stride 2 in the first convolutional layer, instead of the $[11 \times 11]$ filters with stride 4 found in the original AlexNet, and that some hyperparameters of the layers were modified. The main intuition behind this decrease in the input filter size is that smaller filters help to retain more spatial information, thus the net is able to *see* more of the input intrinsic patterns, before applying the following deeper layers.

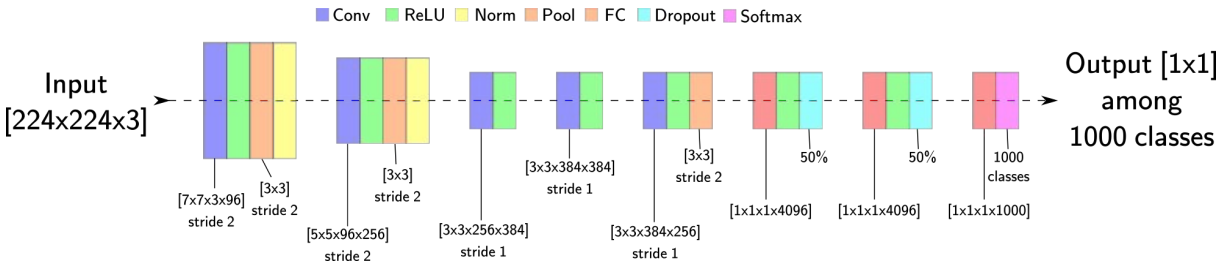


Figure A.3 – Illustration showing the architecture of M. Zeiler and R. Fergus CNN: ZFNet. This CNN was the winner of 2013 ILSVR Challenge with an error rate of 11.2%. ZFNet is an improvement version of 2012 AlexNet, with some modified filter and pooling sizes and strides.

Although the architecture proposed by M. Zeiler and R. Fergus was simply a revision of AlexNet, the truth is that their paper Zeiler and Fergus (2014), on which their classification results and methodologies were explained, is still considered up to date as one of the most important papers on Convolutional Neural Networks. That is so because there Zeiler and Fergus introduced a new method for visualizing CNNs. Instead of showing the activations that a certain image would have after going through the layers of the trained network, they proposed to attach a *Deconvolutional Network* Zeiler et al. (2011) to each one of the layers of the already trained CNN to map the extracted features from each

one of these layer back to the original input image pixel size, by using deconvolutions, unpooling and rectification layers.

This process allow us to visualize which structures in the original image input in the net cause a certain group of neurons to excite. Thus, it is possible to check if the structures that activate each layer in the net do correspond with the classes or types of groups we are aiming to classify or recognize with our net. Being able to check the greatest activations for each class in real pixel is vital to check the discriminative power of the network, as well as to evaluate how well the training procedure went. Figure A.4 shows an example of these deconvolved activations for layer 4 and 5 of the original ZF Net. It is possible to see how the structures of the original images (images on the left, mainly in gray tones) that most activate each class do have a great correlation with the classes that the authors were aiming to classify in the problem (dogs, persons, etc.).

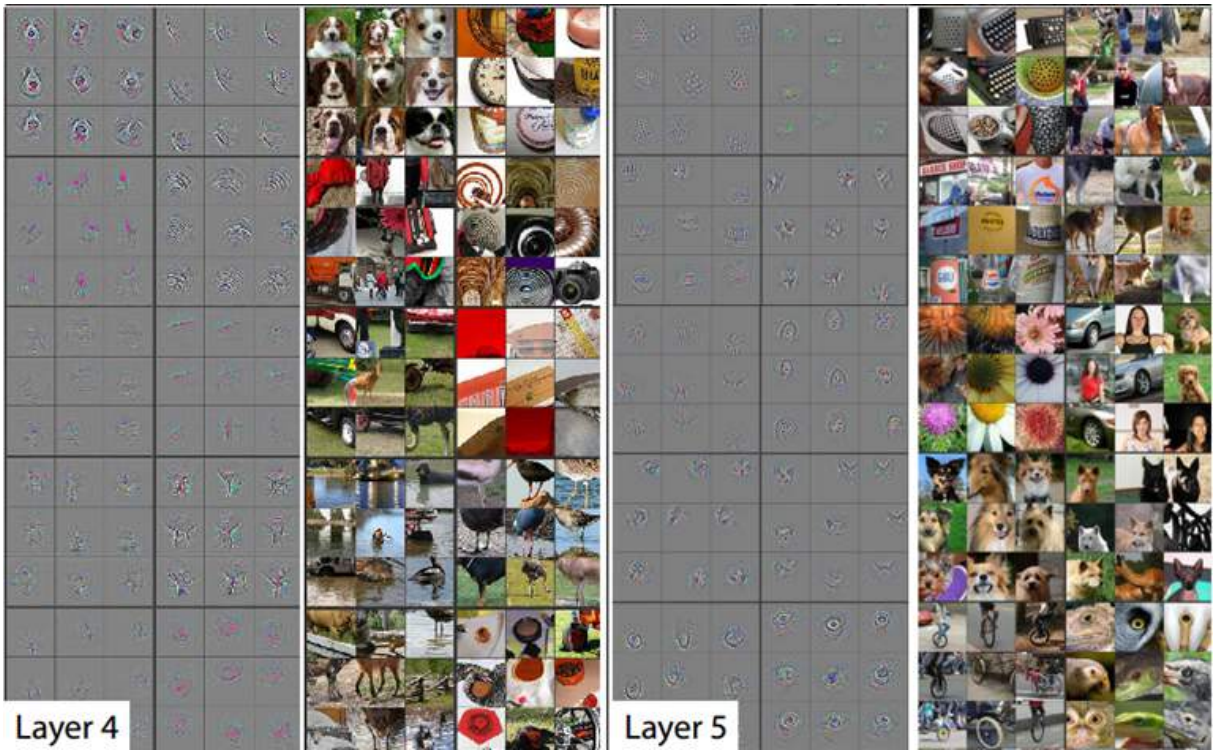


Figure A.4 – Visualizations of some examples of images for different classes used in the ZF–Net and the structures of these images that produce the greatest activations of the net. Source: Zeiler and Fergus (2014).

A.4 VGG Net (2014) – *Simonyan and Zisserman*

VGG Net was presented in the first ILSVR Challenge celebrated in 2014, by its creators Simonyan and Zisserman. Even though they did not win the competition, the VGG Net was the second classified (just below GoogleNet) obtaining a stunning 7.3% error rate: an improvement of almost 4% when compared to the previous year challenge winner, ZF–

Net. In their paper Simonyan and Zisserman (2014) it is possible to check all different structures and combinations that they tested to achieve the final result. Among all of them, the one that performed the best was the architecture using 16 different layers.

What was most surprising about VGG Net success was its simplicity: it is based on a combination of $[3 \times 3]$ stride 1 convolutional layers and $[2 \times 2]$ with stride 2 max–pooling layers, along with Activation and FC layers. Instead of using different sets of filters with higher sizes (like AlexNet or ZF–Net did before them), they decided to consistently use smaller filters in an attempt to decrease the number of total parameters. Apart from that, they argue that a concatenation of 3 $[3 \times 3]$ filters has a receptive field of $[7 \times 7]$, thus this net should be able to extract similar patterns and features as other architectures with bigger filters, but reducing the total amount of parameters.

The final architecture that achieved the best results (and that was presented in the ILSVR challenge) is shown in Figure A.5. It is interesting to notice how the depth of the ConvLayers doubles each time a pooling is applied to the data. This reinforces the idea that the deeper the layer is located inside the CNN, the greater Depth it should have. This concept has been applied to the method for Facies classification proposed in this Section.

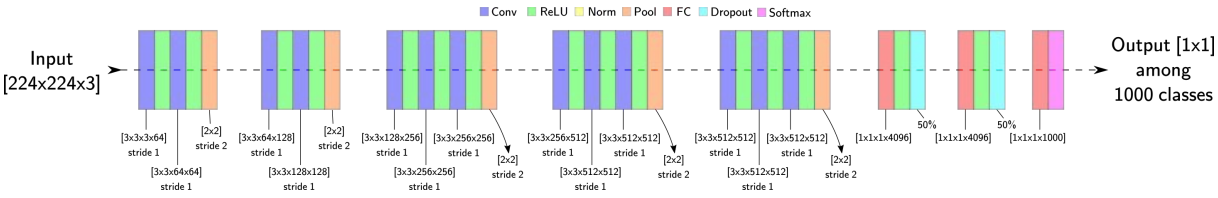


Figure A.5 – Illustration of the full architecture of the VGG–16 Net, which obtained a top–5 classification error rate of 7.3% in the 2013 ILSVR Challenge.

A.5 GoogLeNet (2014) – *Szegedy et al.*

Until the presentation of GoogLeNet in 2014, all proposed CNN architectures were based on Deep stacking of sequential layers. Different combinations of layers produced different features and activations, thus leading to different results; while the improvement on classification and object recognition was thought to be a matter of how these building blocks were combined together. Szegedy et al. changed that landscape in 2014 by introducing the concept of *inception layers*.

An inception layer or inception module can be defined as a group of layers (usually ConvLayers, Pool Layers and ReLU layers) that are gathered in a way such that their operations are applied in parallel. This means that different operations (convolutions, maxpooling, etc.) are applied to the same input data, which comes from previous layers, and its outputs are concatenated at the end of the module so that they can be used in further (inception or common) layers.

Figure A.6 shows an example of an inception module used in the original version of GoogLeNet. In this case the inception module is formed by 4 different parallel groups: a) $[1 \times 1]$ ConvLayer; b) a $[1 \times 1]$ followed by a $[3 \times 3]$ ConvLayers; c) a $[1 \times 1]$ followed by a $[5 \times 5]$ ConvLayers; and d) a $[3 \times 3]$ max–pooling layer followed by a $[1 \times 1]$ ConvLayer. At the end of the module all the responses obtained at each one of these 4 *branches* are concatenated[2] so that it can be used in any following layers (or inception modules).

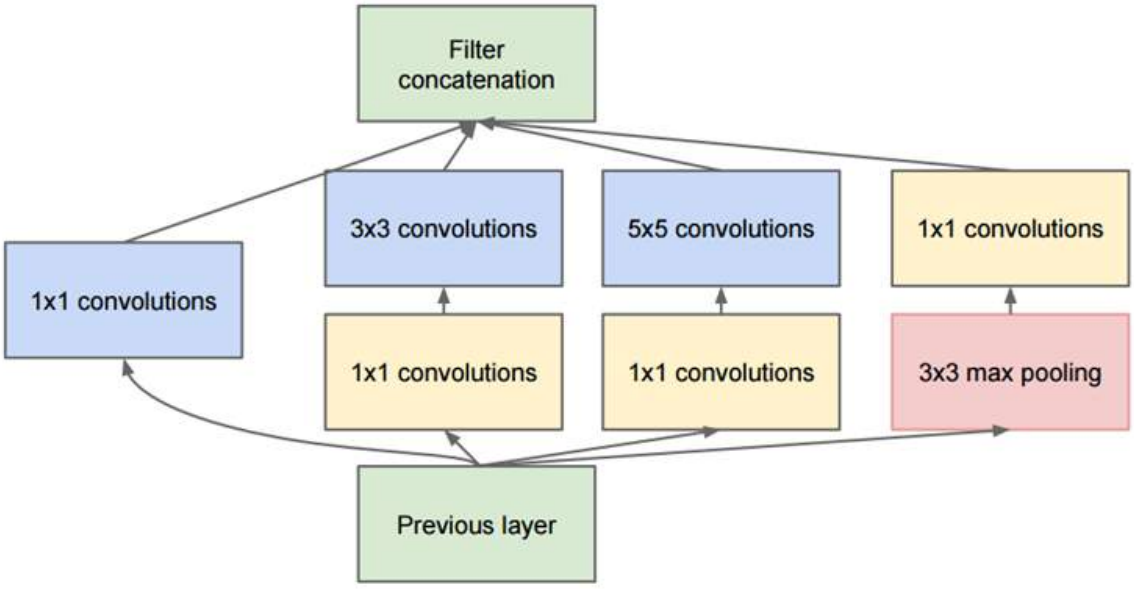


Figure A.6 – Illustration of an inception layer. In this case this layer is formed by 4 parallel operations: a) $[1 \times 1]$ ConvLayer; b) a $[1 \times 1]$ followed by a $[3 \times 3]$ ConvLayers; c) a $[1 \times 1]$ followed by a $[5 \times 5]$ ConvLayers; and d) a $[3 \times 3]$ max–pooling layer followed by a $[1 \times 1]$ ConvLayer. The results of each one of these groups of layers are then concatenated and used in the following structures of the net. Source: Szegedy et al. (2015).

The main purpose when using different filter sizes for the same input data coming from the previous layer is that each one of them might enhance different features of that data. The $[1 \times 1]$ convolutional layers act as dimension reduction units, where the size of the input data is not modified in width, height or number of channels, but its depth its reduced. That way it is possible to obtain a number of parameters so that the final concatenated matrix can be feasibly used in the training process without consuming too many resources.

The first version of GoogLeNet, which achieved a total error rate of 6.7%, was formed by a total of 22 modules/groups of layers, which were built up with a total of 154 single layers. Figure A.7 shows the full diagram of this version of GoogLeNet.

The main highlights of GoogLeNet were:

[2]When data with different sizes must be concatenated, the most common approach is to pad with zeros the smaller data so that it becomes the same size as the data with the biggest size.

1. Improvement in the classification and object recognition error, which was reduced down to 6.7%.

2. Most Deep architecture created up to that date with over 150 layers in total.

3. Use of Average PoolLayers instead of FC layers to create dense layers and activations. This feature allows the net to save a huge number of parameters (i.e., a total of 12× less parameters than AlexNet).

4. First CNN architecture ever to use inception models as parallel processing units, showing that complex non–stacking structures were not only possible, but also provided great results when dealing with image recognition and classification.

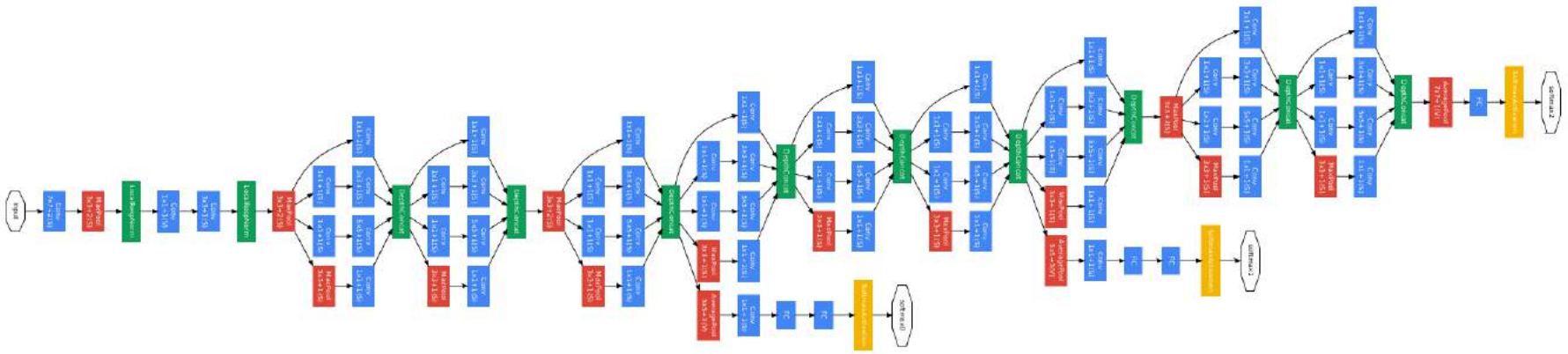


Figure A.7 – Complete diagram of the first GoogLeNet version, which won the ILSVR Challenge in 2014. This was the first CNN architecture to use inception modules (as shown in Figure A.6) which allowed several layers and operators to be applied parallelly during the data stream. The input data starts on the left, while there exists three different sotfmax layers (yellow squares) that obtain three different predictions, which act as the net outputs (blank octagons), regarding the input data sample class. Blue squares represent convolutional and fully–connected layers, red squares represent max–pool layers while green squares represent the concatenation of activation maps along depth. This concatenation of maps is the essential feature intorduced by inception modules: they allow the net to extract different properties and features from the same data at different levels (using different sizes of filters kernels, or different pooling operations) and then concatenate them into a single activation map. Source: Szegedy et al. (2015)

A.6 Microsoft ResNet (2015) – *He et al.*

The ILSVR Challenge of 2015 represented a major step forward in the image processing and CNN world because it was the first time that the error rate of an artificial system overcame the average accuracy of human beings. The winner of that year challenge was a CNN architecture called *ResNet*, developed by the Microsoft AI research time He et al. (2016). This net achieved an incredible 3.6% error rate, much better than the average humans error rate, which ranges from 5 to 10% depending on the level of expertise of the user.

The main idea behind ResNet was that instead of making the net optimize the activation functions of singular data after each operational block, it would be easier for the net to optimize the differential activation functions that each block had to add to the previous layer result. Thus, let us image some data x coming from a previous block in the net which suffers a certain transformation f due to the activation function of a block (composed by convolutional, pooling and or other types of layers), after which the data can be represented as $f(x)$; at the end of the Residual block the original data x is added to the transformed data, so that the output data of the ResBlock is $f(x) + x$ instead of x. Figure A.8 illustrates the concept of a single Residual Block.

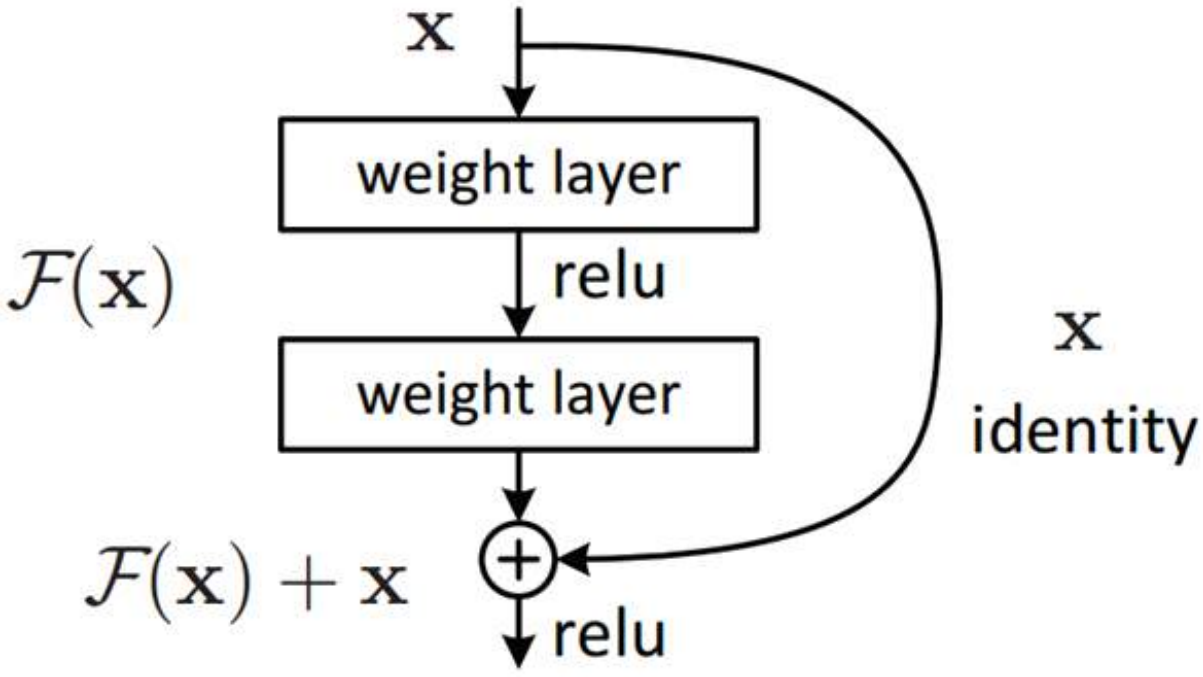


Figure A.8 – Illustration of the concept of a Residual Block. On each one of these blocks the data coming from the previous block x goes through some layers or blocks, being transformed into $f(x)$; at the end of the block, this transformed data is added to the original input data, so that the output of the block is $f(x) + x$. Source: He et al. (2016).

The complete architecture of the first version of ResNet is shown in Figure A.9. As it

can be seen in the figure, the net was formed by 50 Residual Blocks, each one composed by 3 groups of ConvLayer + NormLayer + ReLU. He et al. argue in their paper He et al. (2016) that it is easier for the net to optimize the residual mapping than to optimize the original, unreferenced mapping. There they also show the different results that they obtained for different tested architectures: a plain network inspired in VGG16 with more layers and ResNets with different sizes. He et al. showed that simply stacking layers on plain architectures did not necessarily mean that the error rate would decrease, while ResNets with higher number of layers did perform better than ResNets with fewer of them.

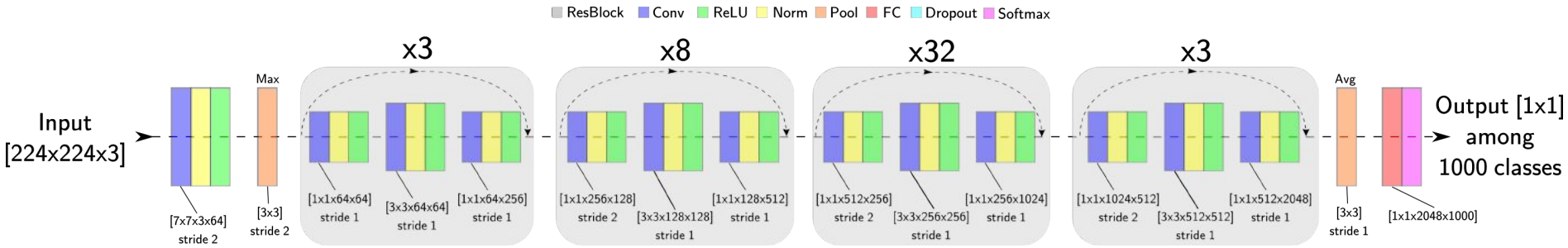


Figure A.9 – Complete diagram of the first ResNet version, which won the ILSVR Challenge in 2015. This was the first CNN architecture to use *residual blocks* and to overcome the human average classification rate, obtaining a top–5 error rate of 3.6%. Source: He et al. (2016)

APPENDIX B

Lithofacies Classification Results

The following figures display the 60 blocks with the highest prediction error, sorted from worse (the whole block was misclassified) to better (only some samples of the block were misclassified). The first three columns display the (a) depth, (b) ultrasonic data and (c) resistivity data of that interval, while the forth and fifth columns show the (d) target facies of that interval (made by geologists based not only on the borehole image data presented here, but also in raw logs, core analysis and seismic data) and the (e) prediction made by the model proposed in Section 5.

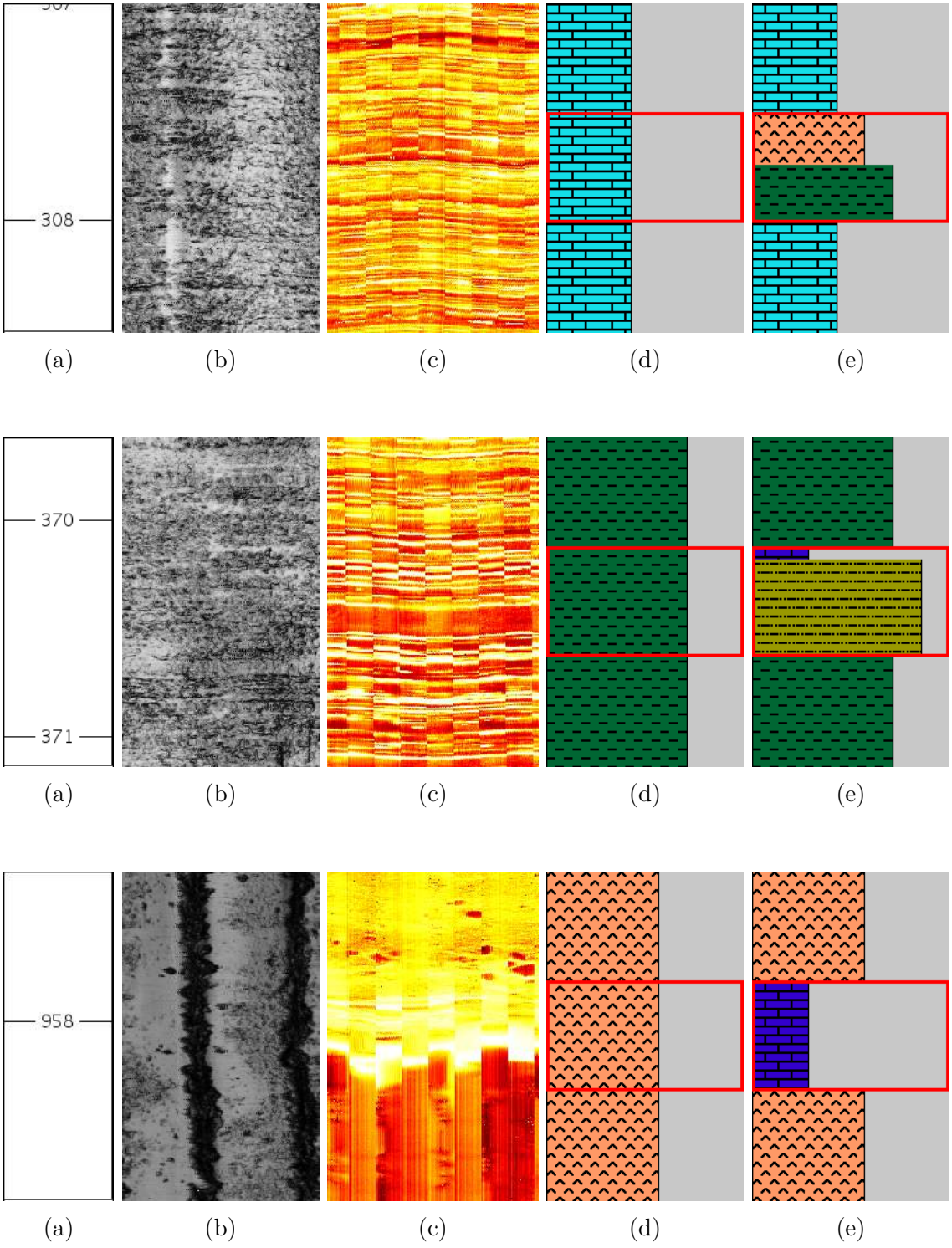


Figure B.3 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

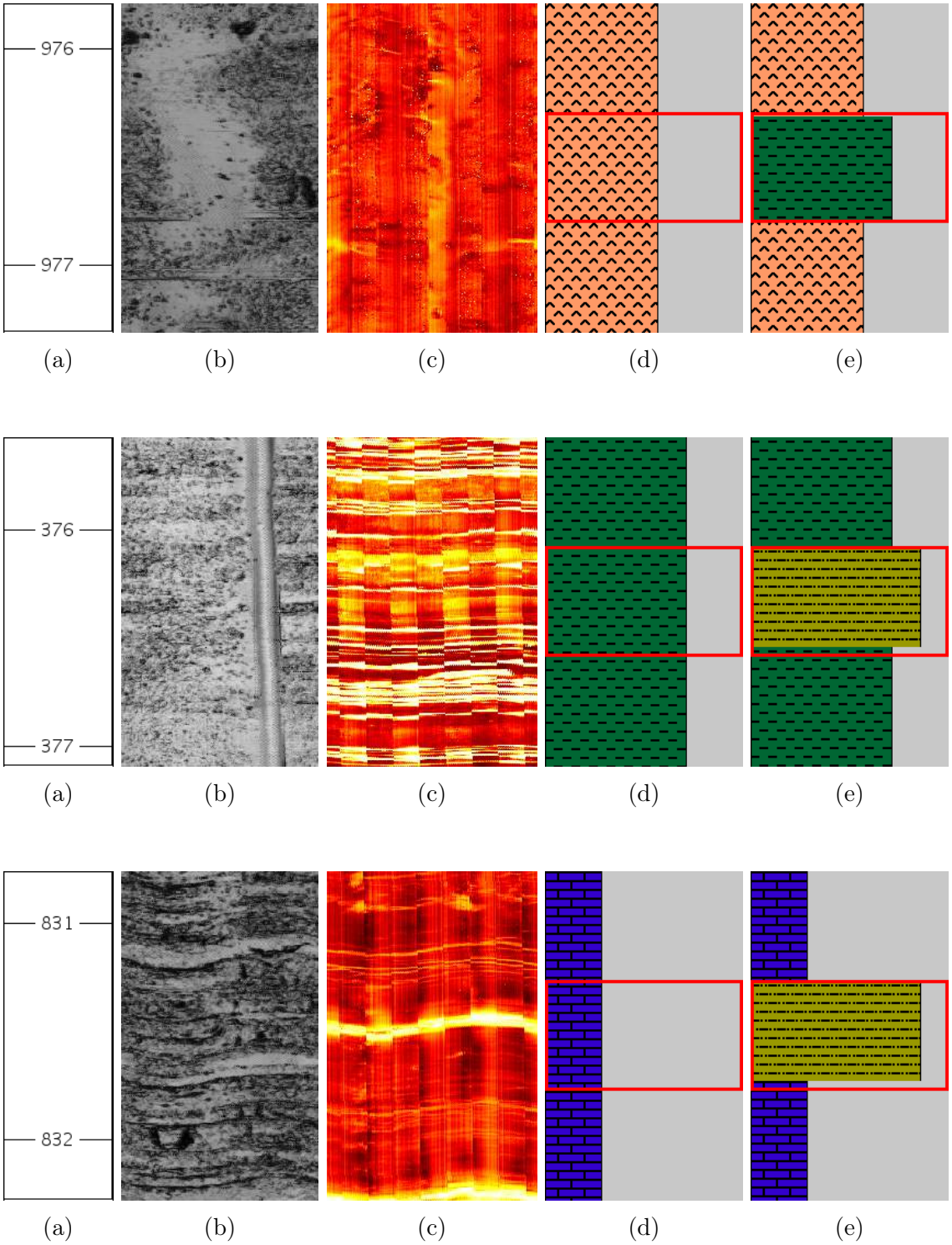
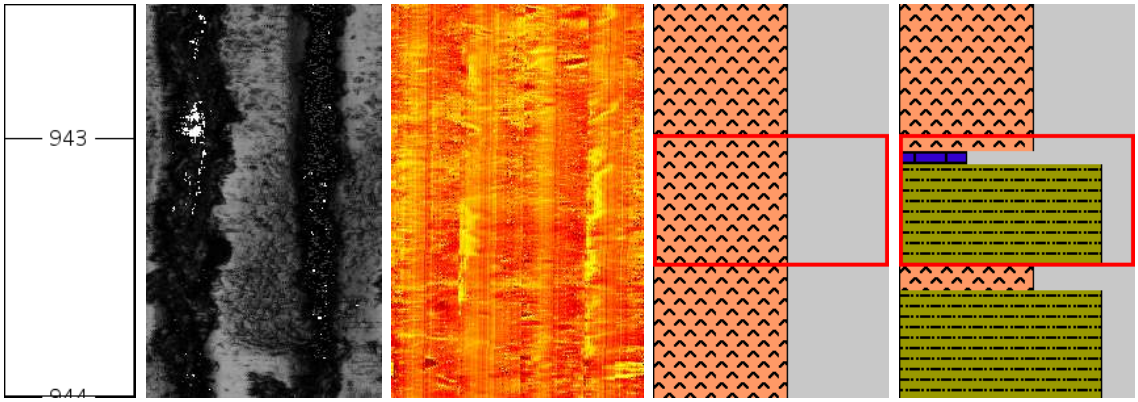
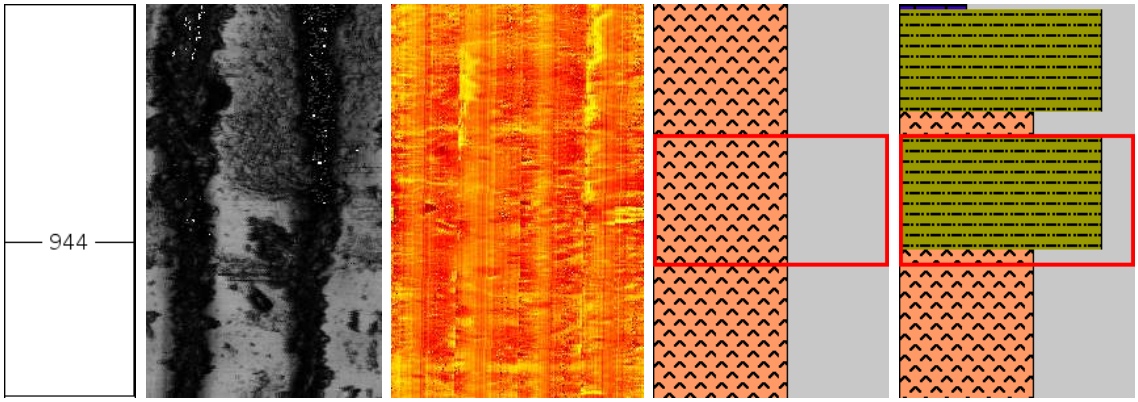


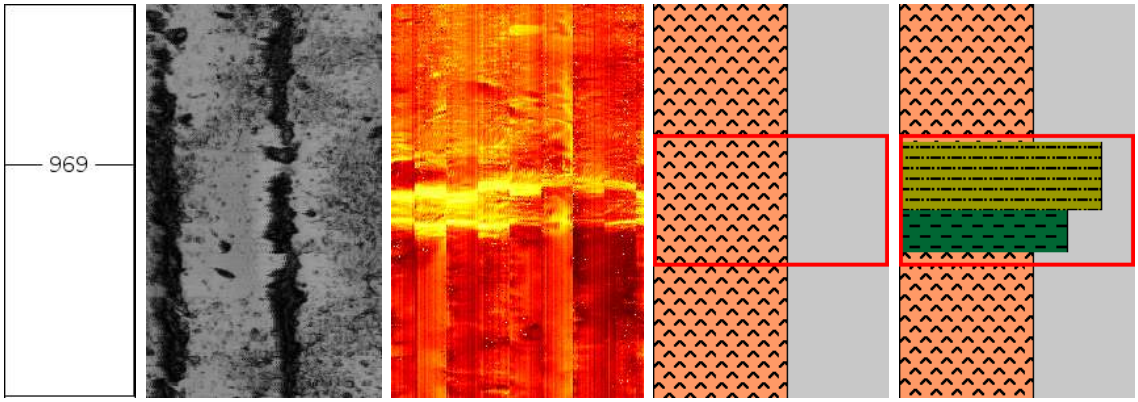
Figure B.6 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.



(a) (b) (c) (d) (e)



(a) (b) (c) (d) (e)



(a) (b) (c) (d) (e)

Figure B.9 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

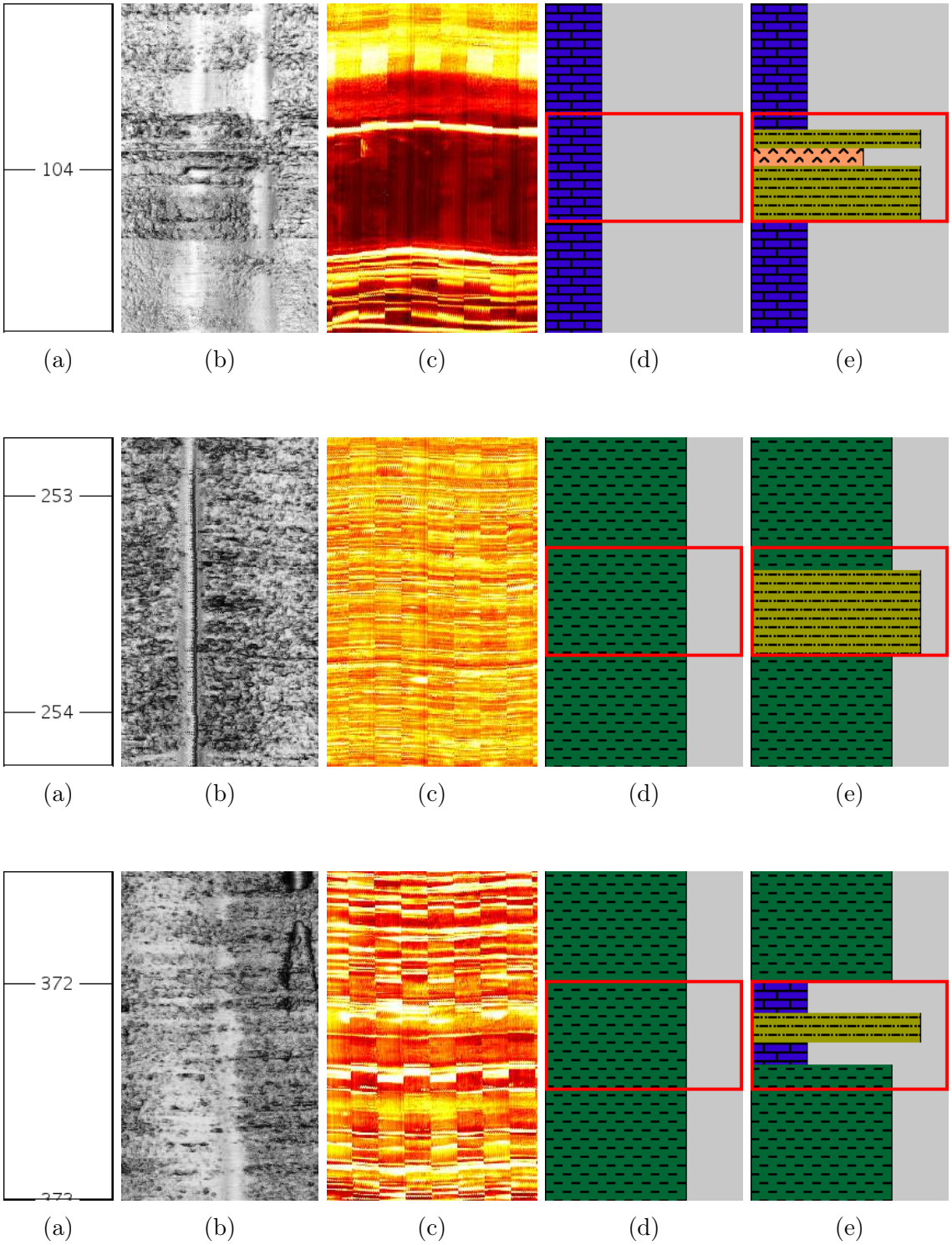


Figure B.12 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

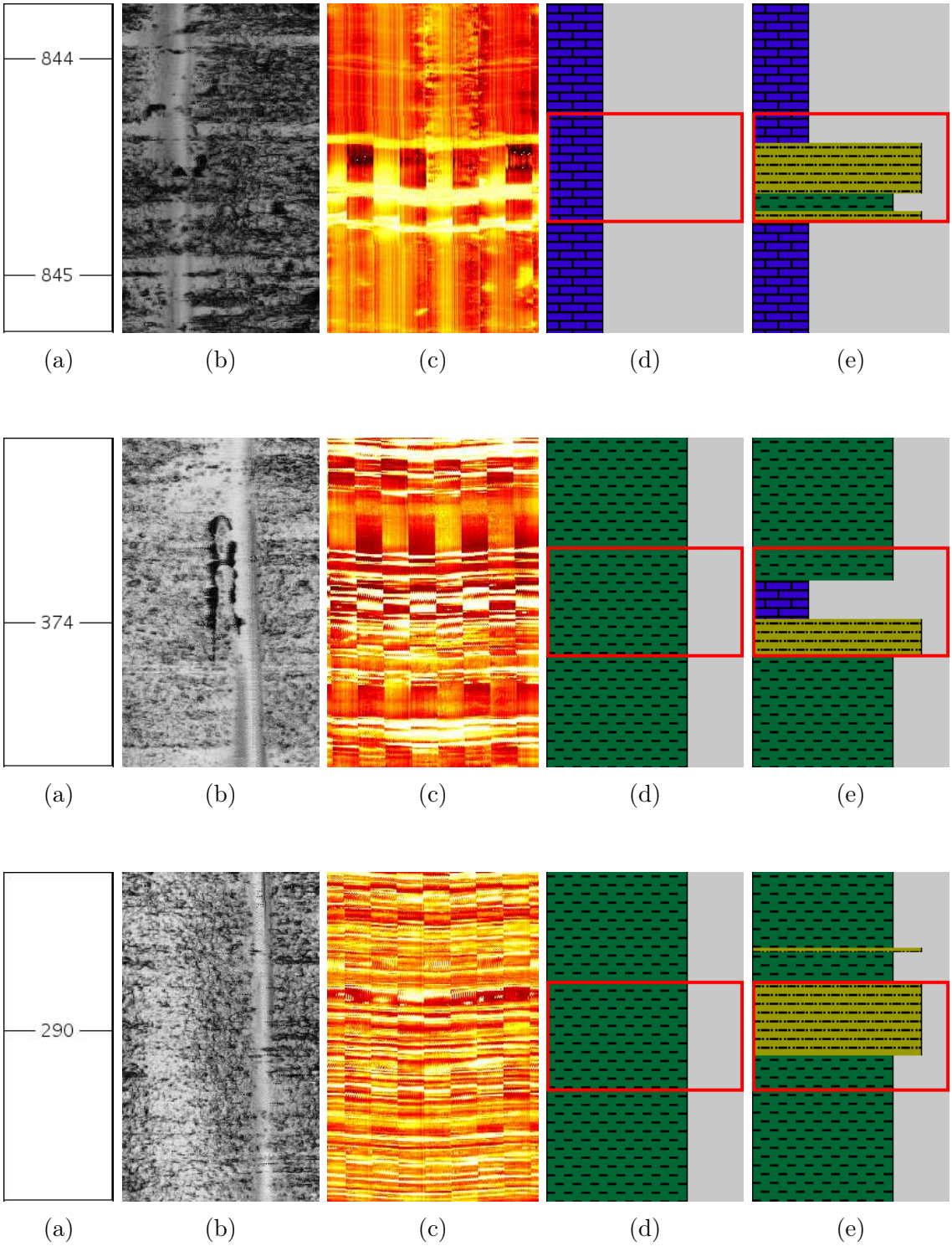


Figure B.15 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

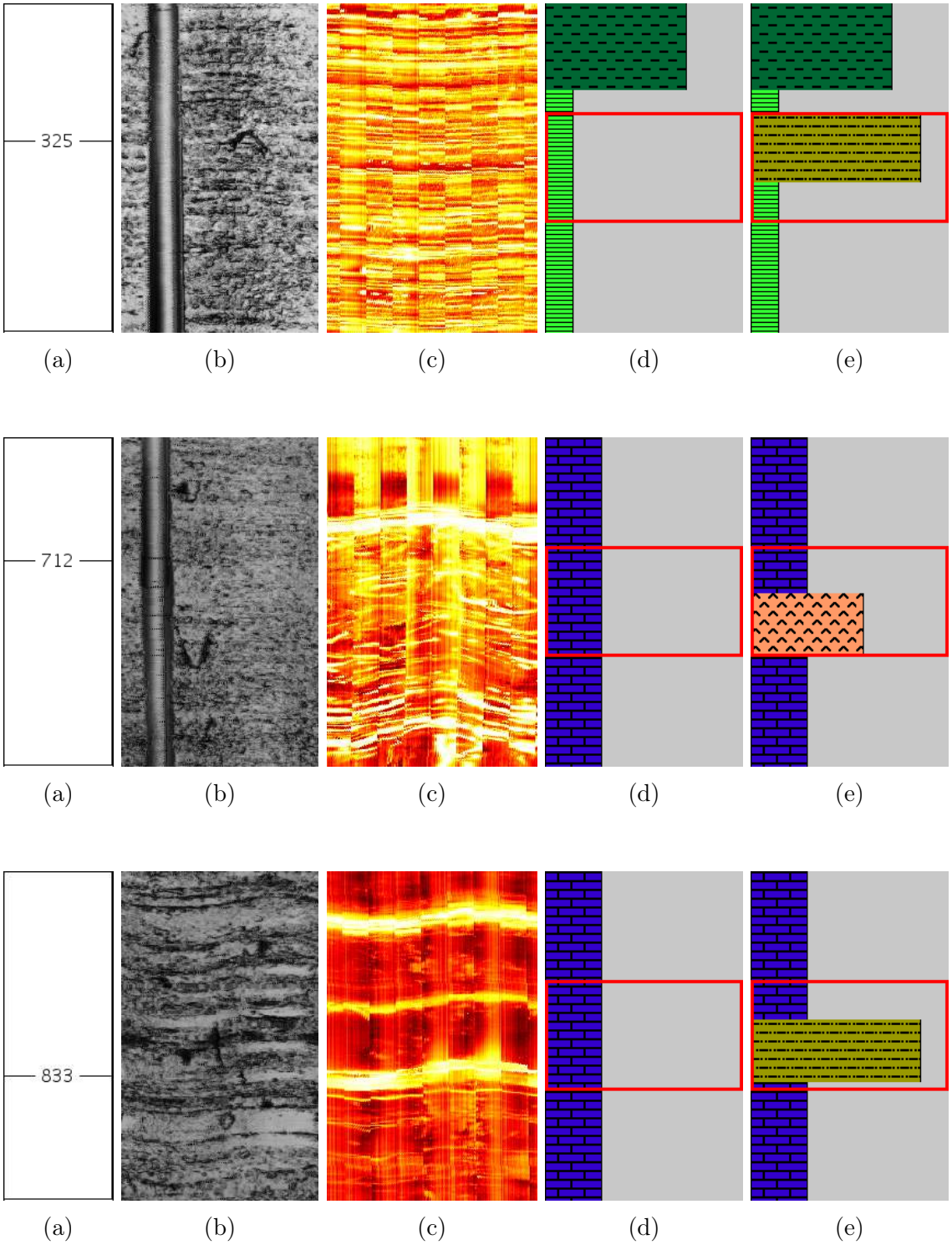


Figure B.18 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

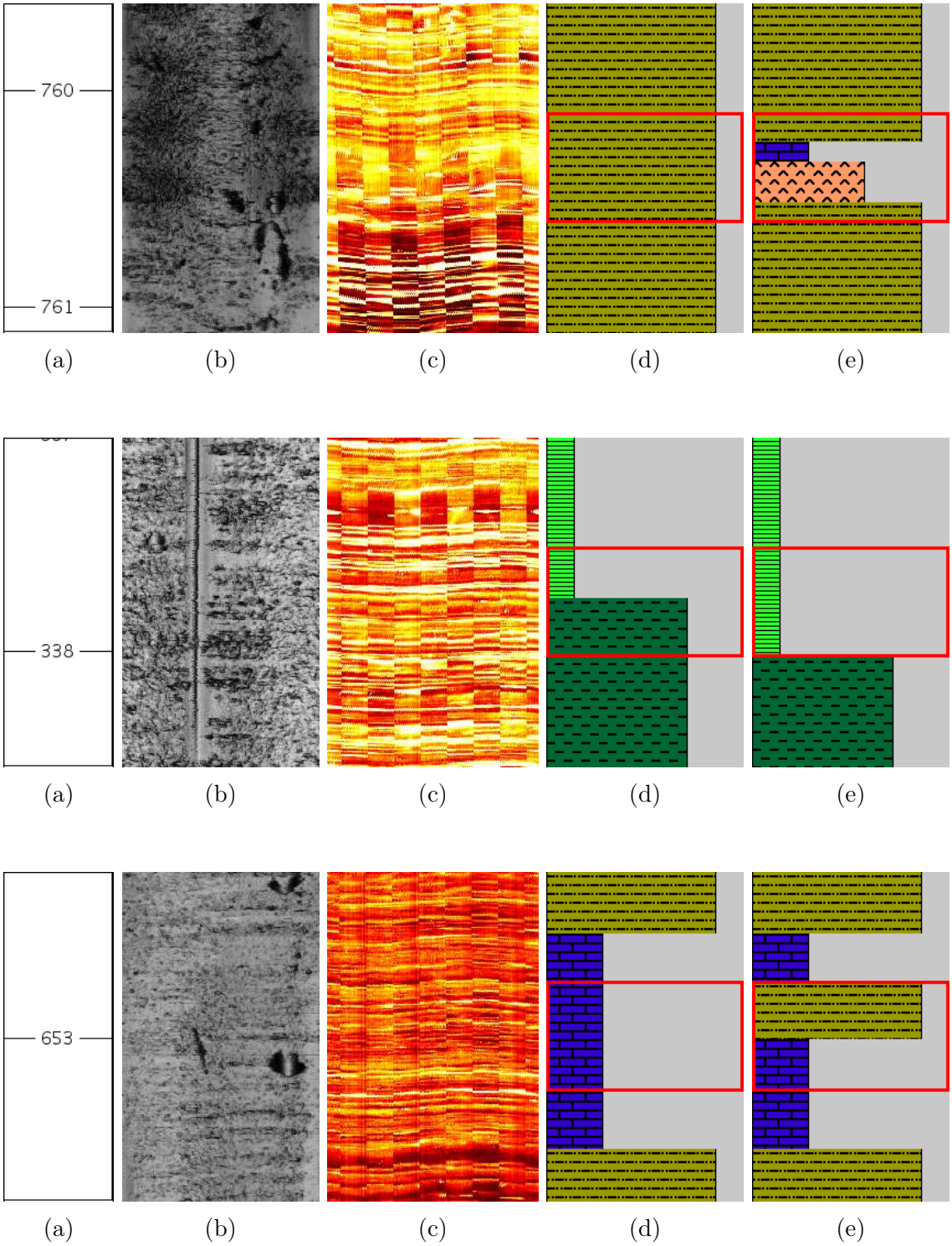


Figure B.21 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

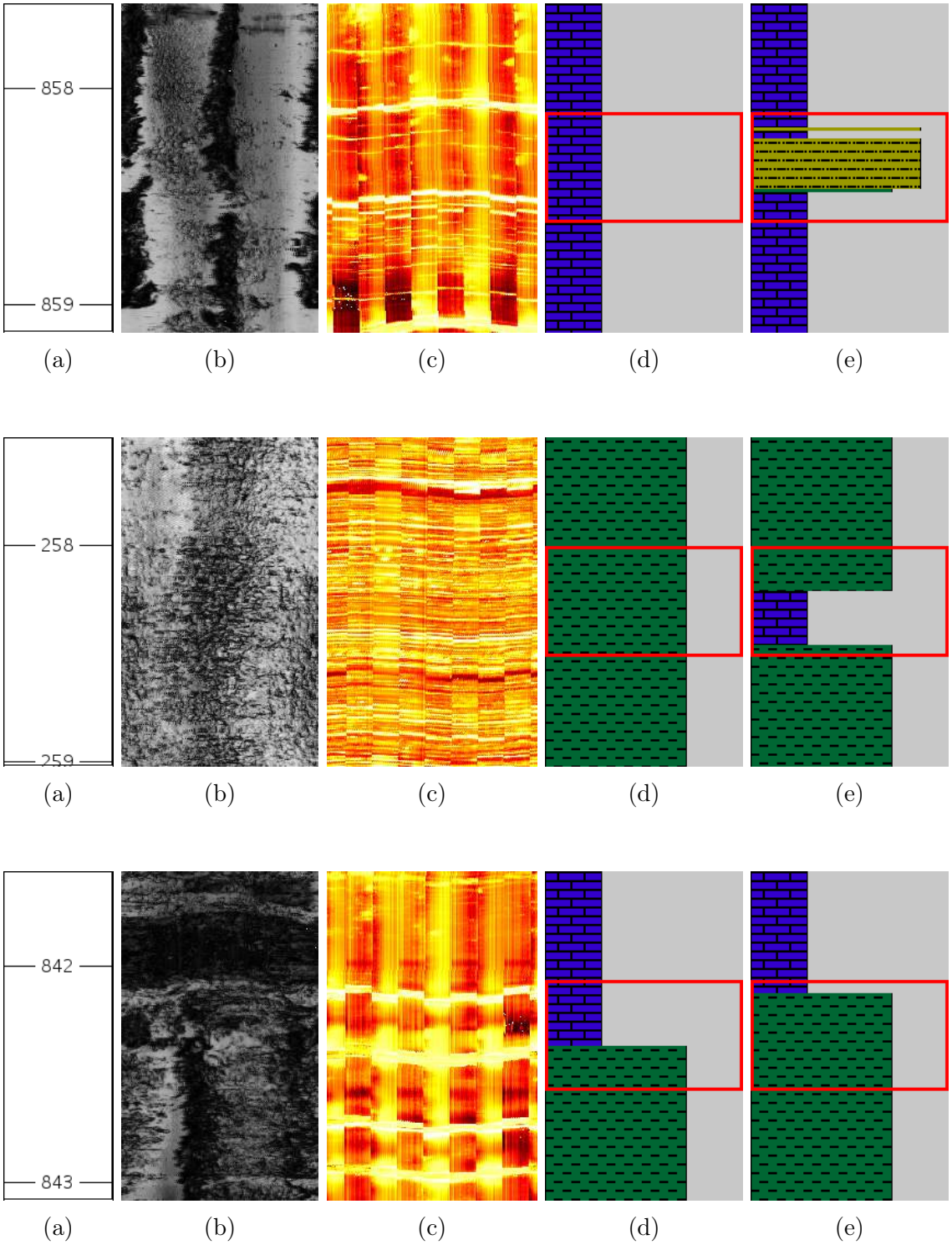


Figure B.24 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

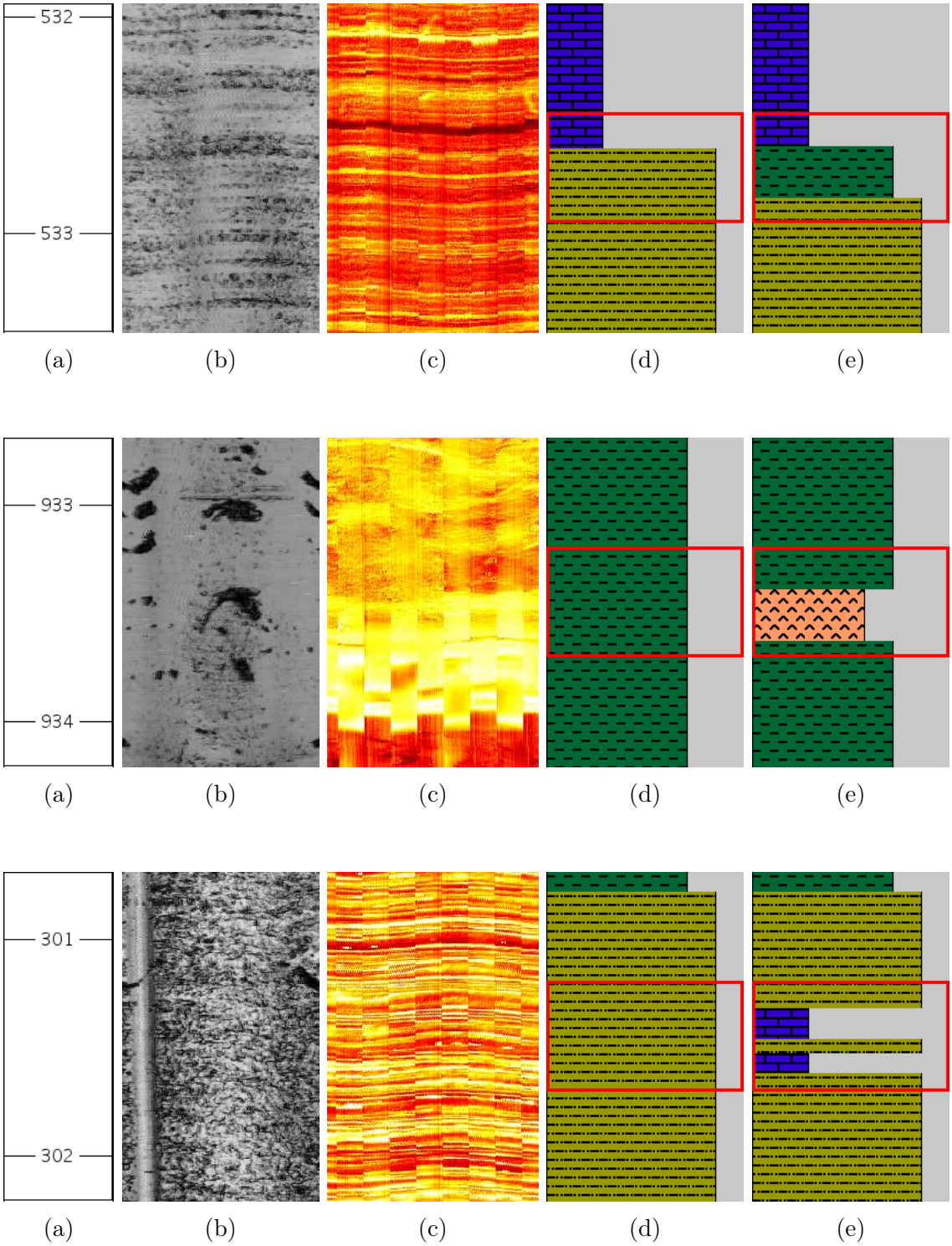


Figure B.27 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

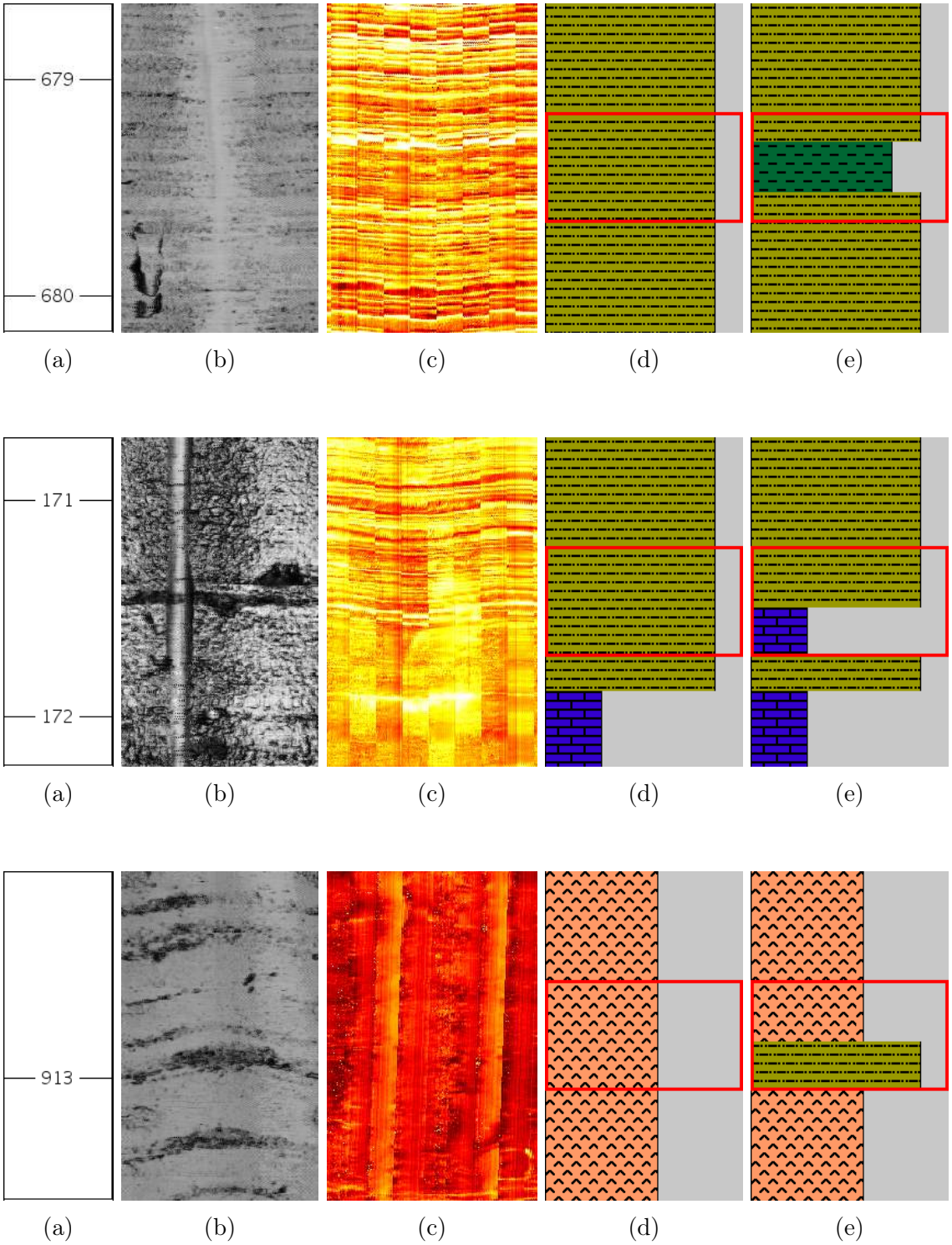


Figure B.30 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

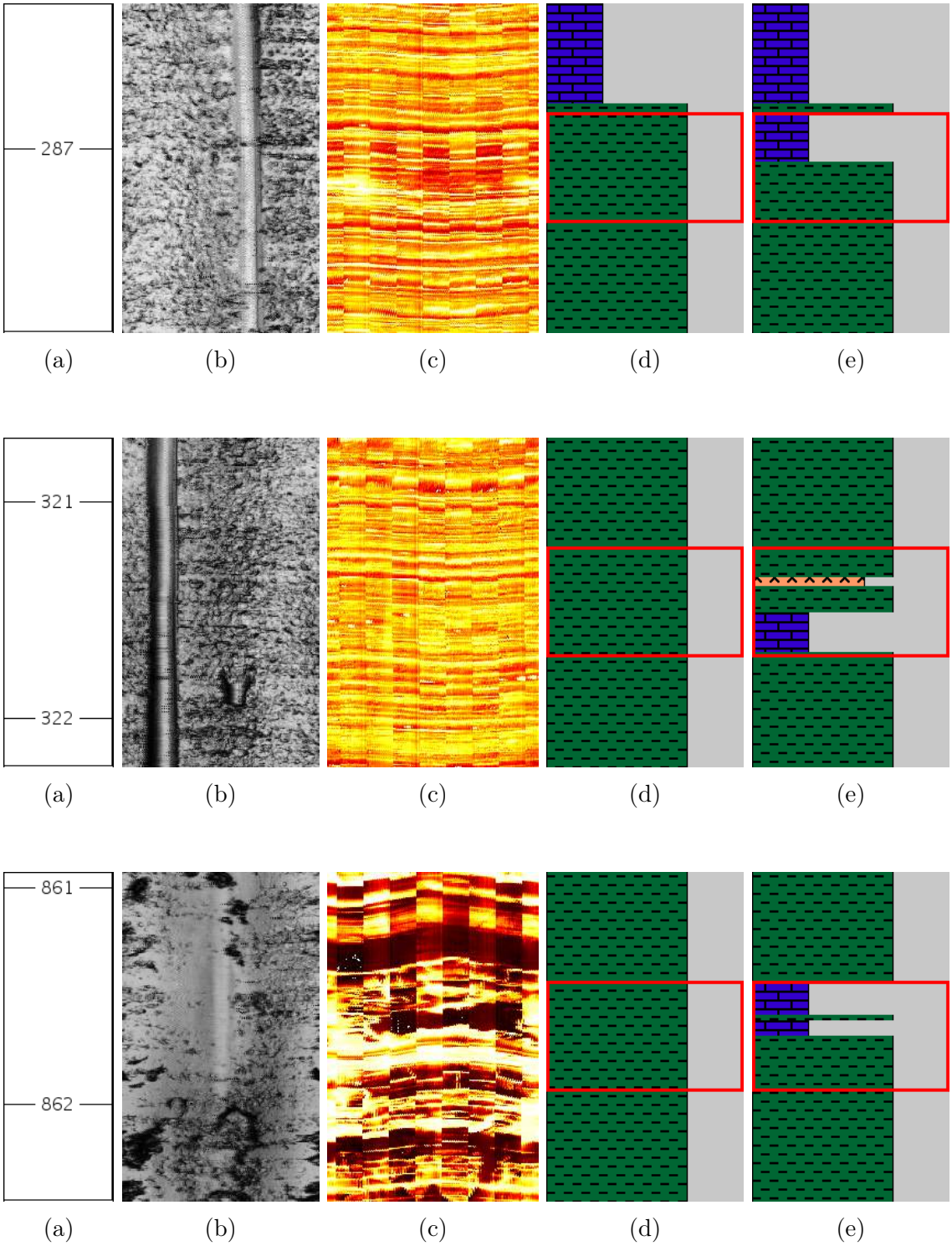


Figure B.33 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

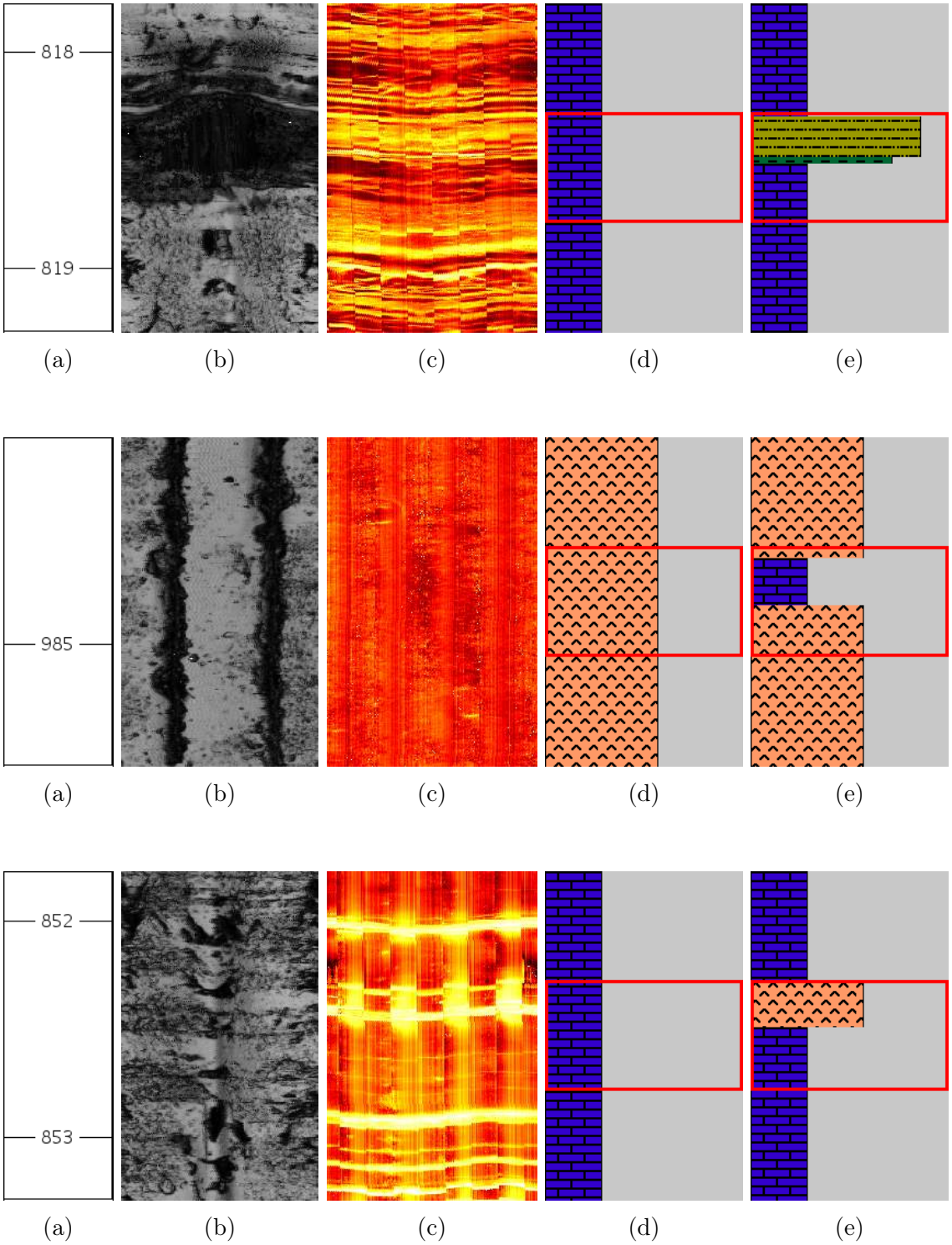


Figure B.36 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

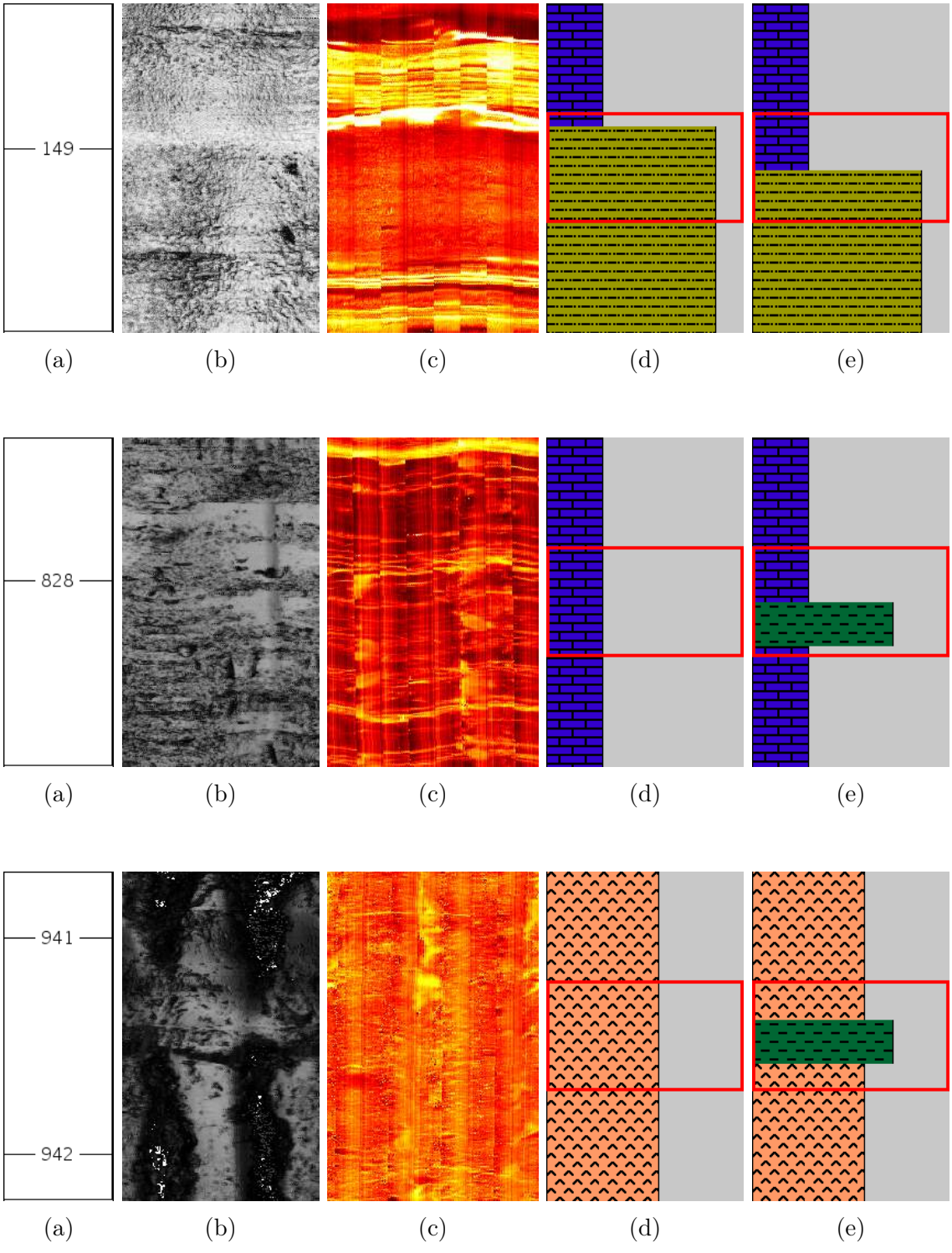


Figure B.39 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

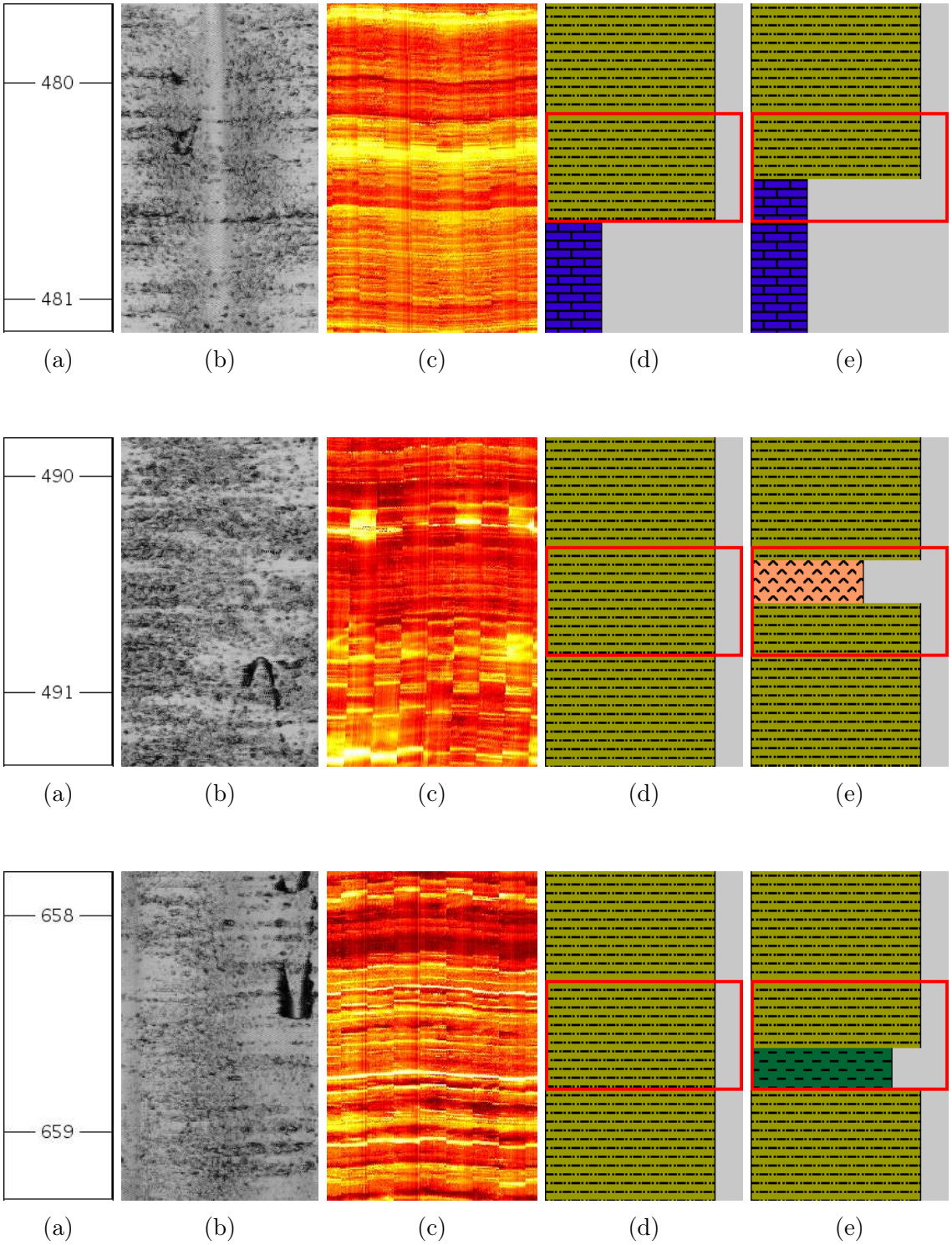


Figure B.42 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

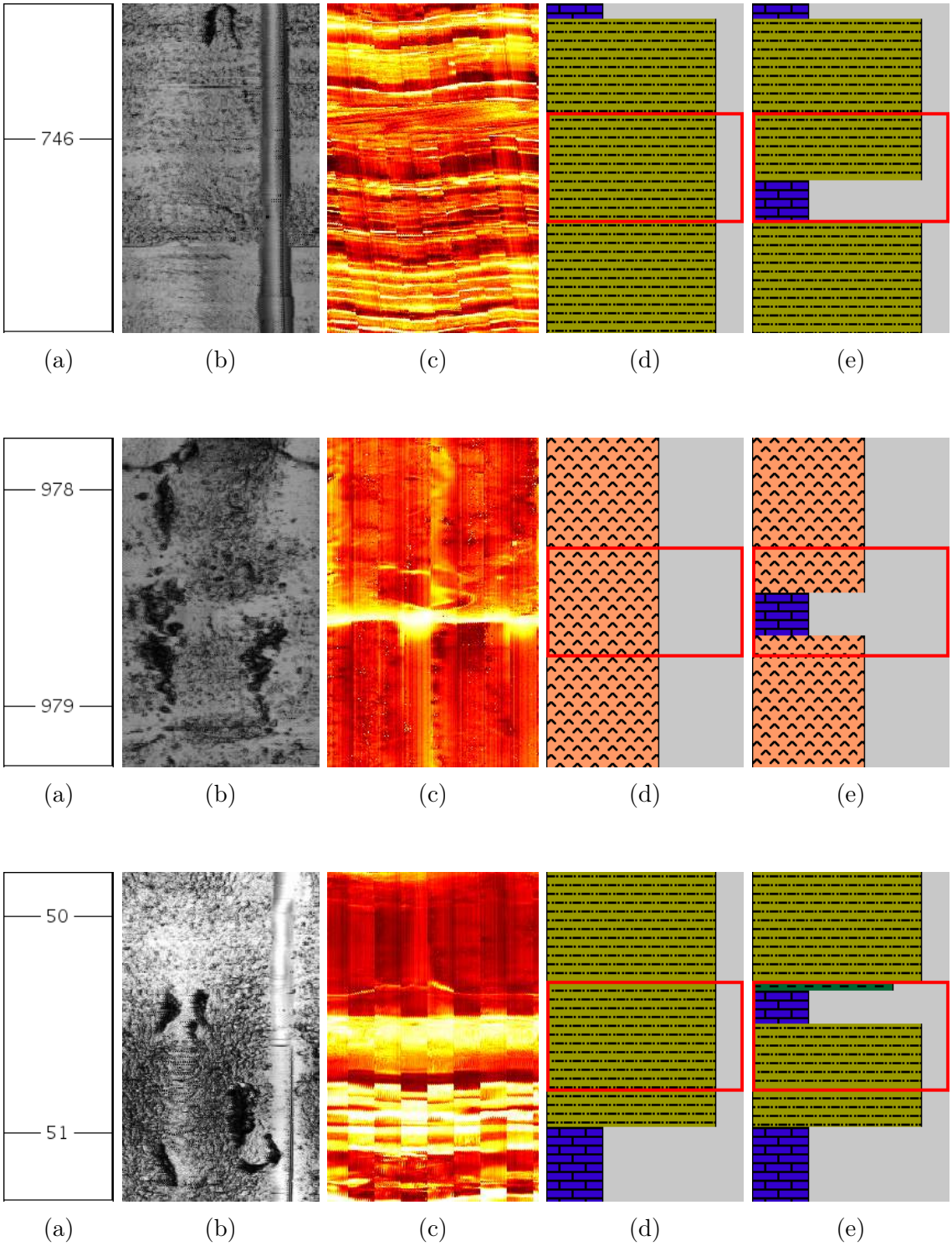


Figure B.45 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

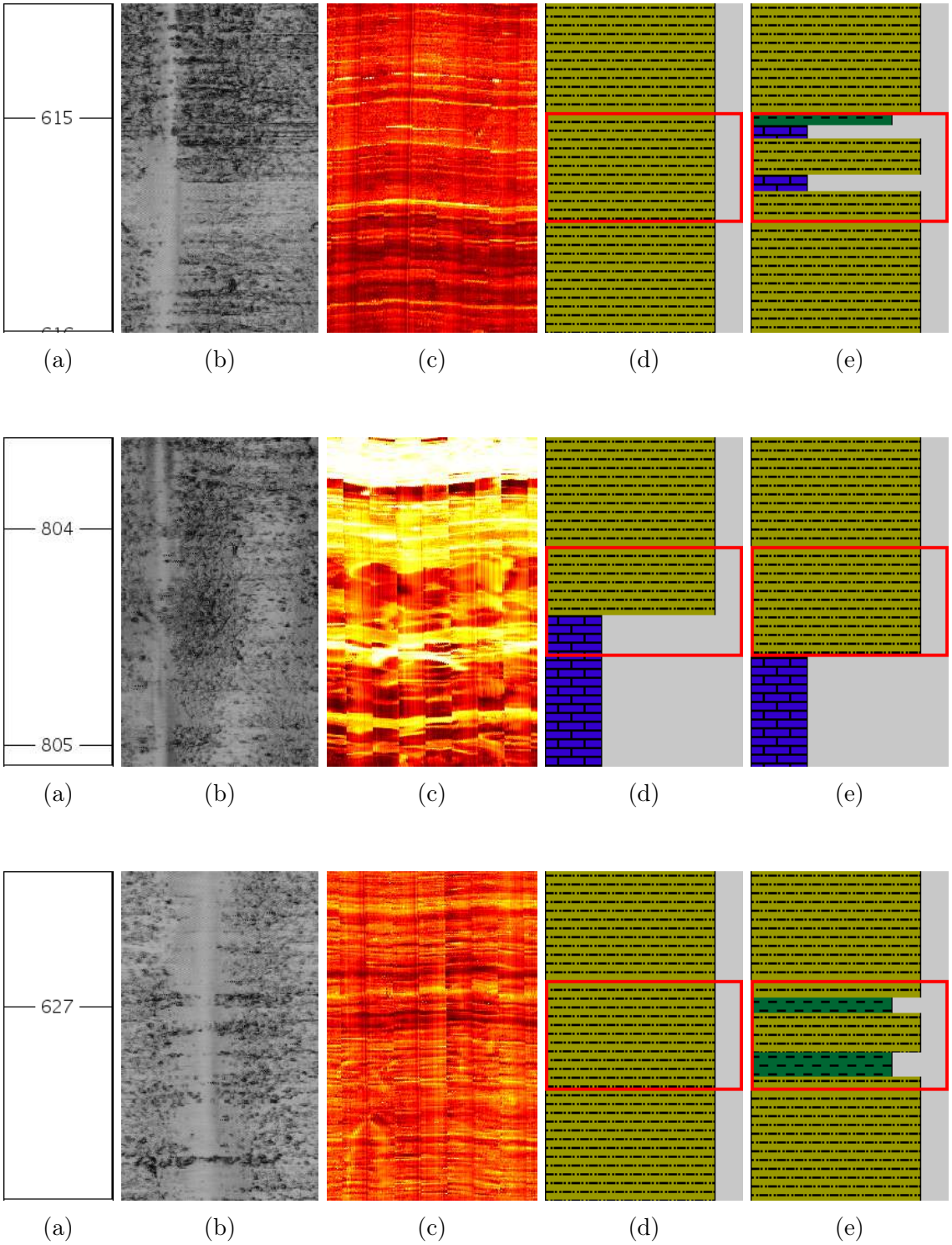


Figure B.48 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

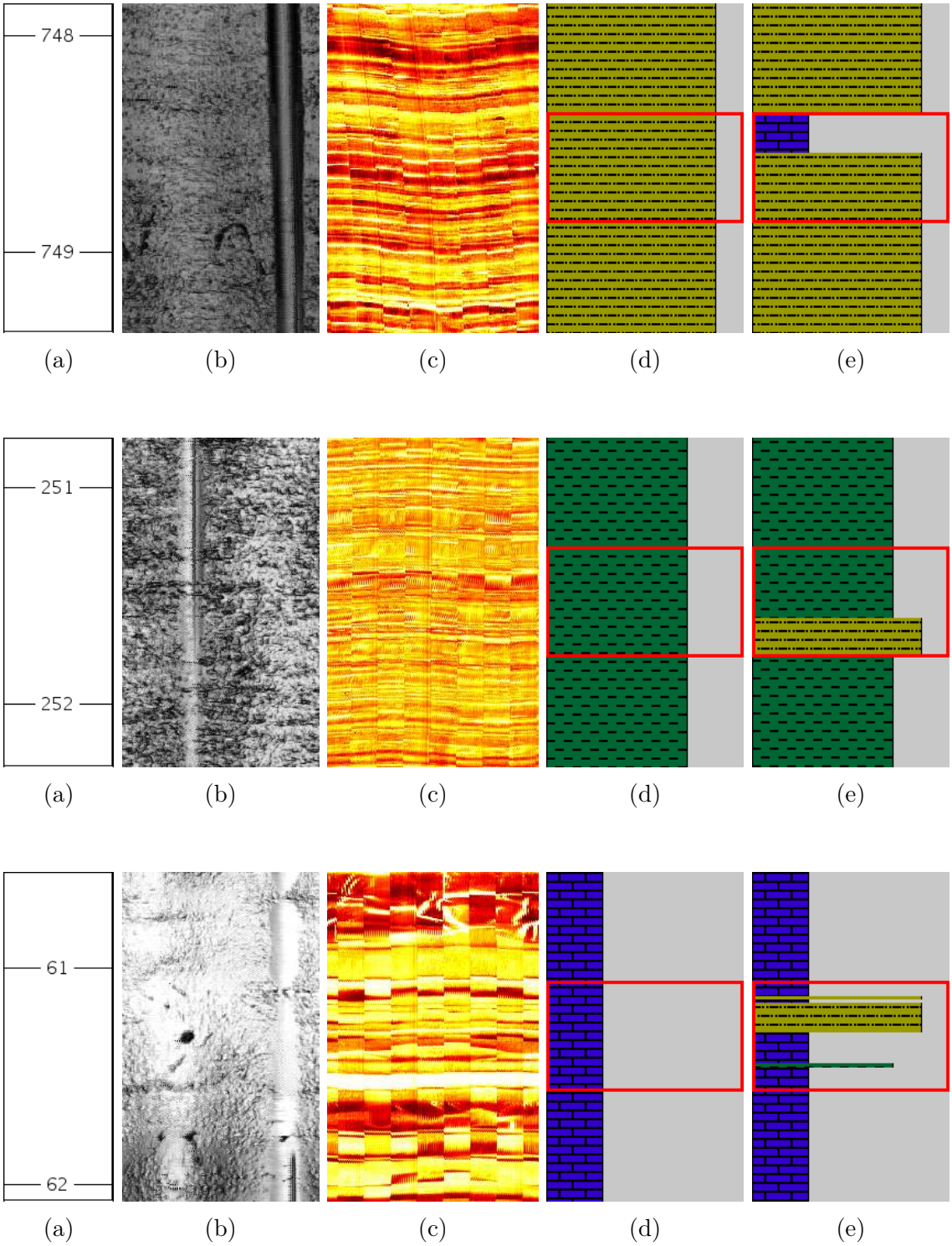


Figure B.51 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

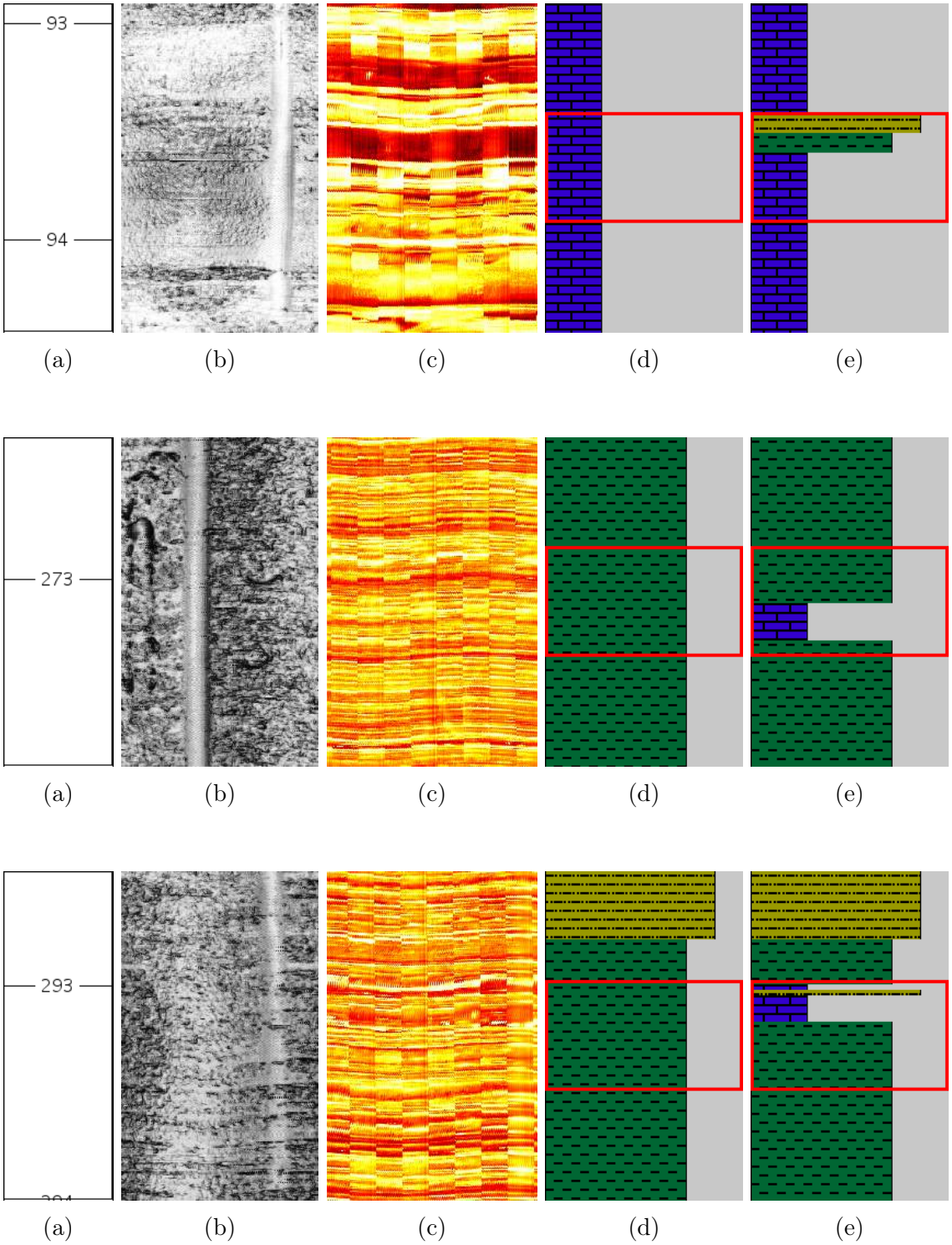


Figure B.54 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

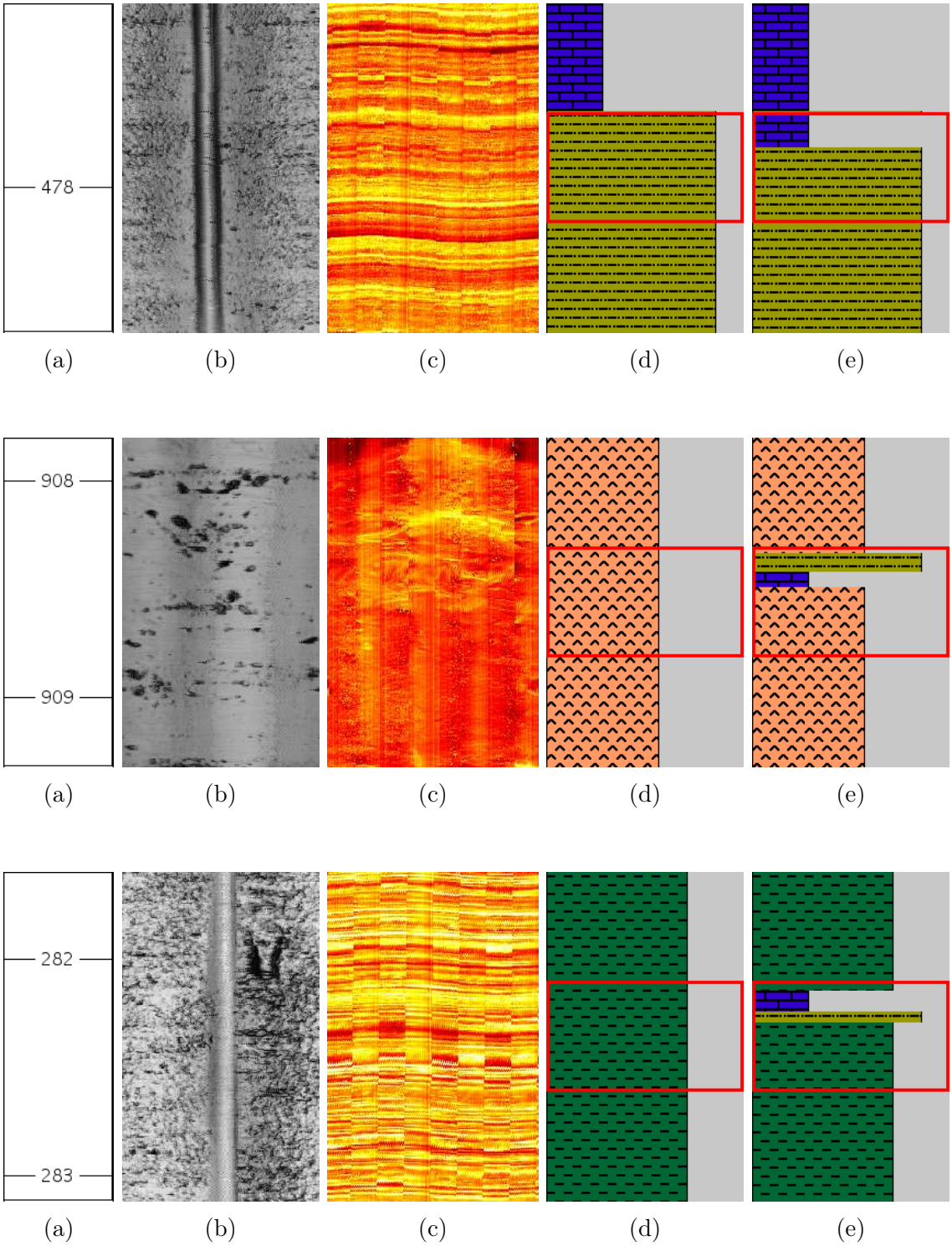


Figure B.57 – a) Well depth reference of the analyzed interval and respective b) ultrasonic and c) microresistivity borehole image data, d) lithofacies target (made by geologists) and e) classification prediction using the proposed method.

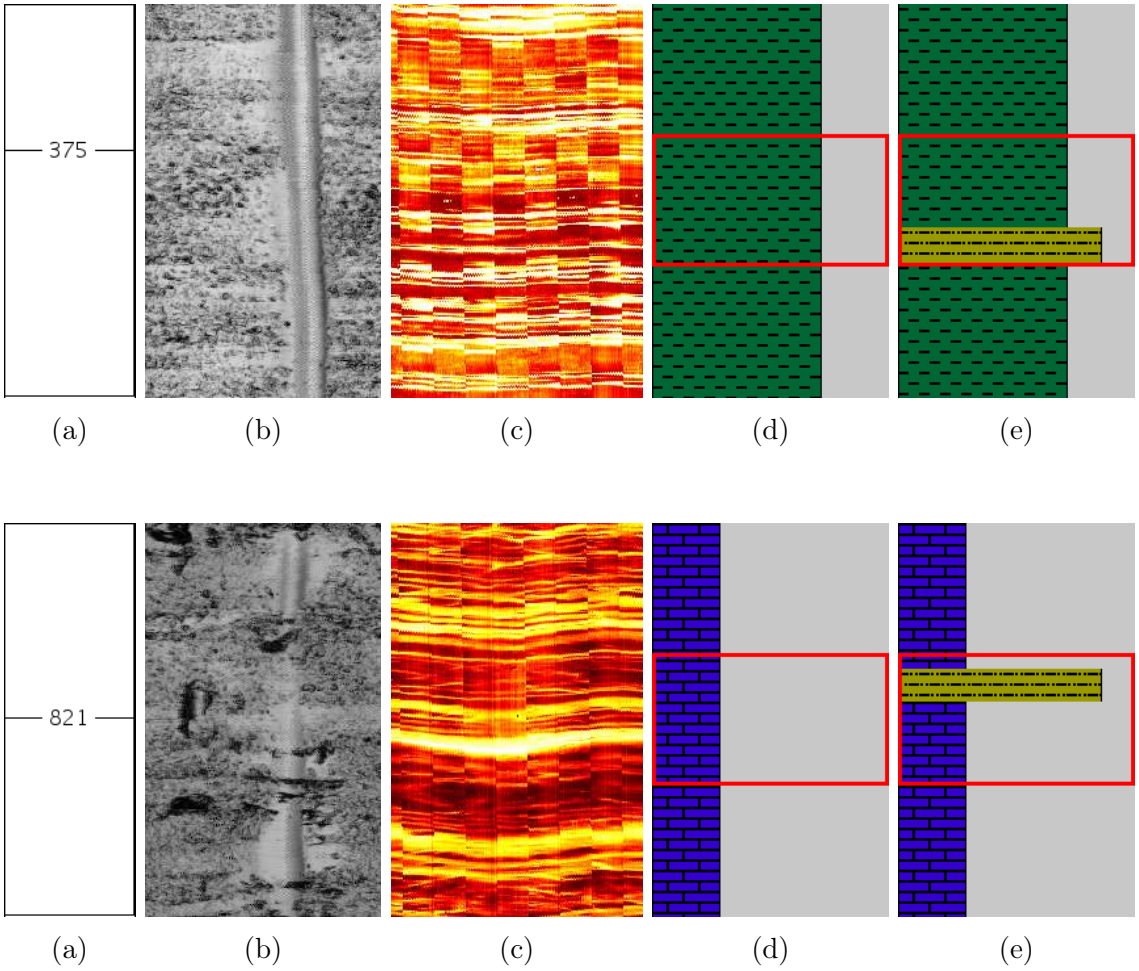


Figure B.59 – a) Well Depth Reference of the analyzed interval. b) Ultrasonic and c) Microresistivity Borehole Image Data at the specified interval. d) Target classification of the Sedimentary Facies of the specified interval (made by Geologists). e) Classification Prediction using the proposed method.