Computers & Geosciences 193 (2024) 105730

Contents lists available at ScienceDirect

Computers and Geosciences

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

Review article

Rock-type classification: A (critical) machine-learning perspective

Pedro Ribeiro Mendes Júnior [a,*], Soroor Salavati [a], Oscar Linares [a], Maiara Moreira Gonçalves [b], Marcelo Ferreira Zampieri [a], Vitor Hugo de Sousa Ferreira [b], Manuel Castro [a], Rafael de Oliveira Werneck [a], Renato Moura [a], Elayne Morais [a], Ahmed Esmin [a,c], Leopoldo Lusquino Filho [d], Denis José Schiozer [e], Alexandre Ferreira [a], Alessandra Davólio [b], Anderson Rocha [a]

[a] Universidade Estadual de Campinas (UNICAMP), Instituto de Computação (IC), Artificial Intelligence Lab., Recod.ai, Av. Albert Einstein, 1251, Cidade Universitária, 13083-852, Campinas, São Paulo, Brazil
[b] UNICAMP, Centro de Estudos de Petróleo (CEPETRO), Rua Cora Coralina, 350, Cidade Universitária, 13083-896, Campinas, São Paulo, Brazil
[c] Universidade Federal de Lavras (UFLA), Departamento de Ciência da Computação (DCC), Av. Central, 3037, Centro, 37200-900, Lavras, Minas Gerais, Brazil
[d] Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia (ICT), Automation and Integrated Systems Group (GASI), Av. Três de Março, 511, Alto da Boa Vista, 18087-180, Sorocaba, São Paulo, Brazil
[e] UNICAMP, Faculdade de Engenharia Mecânica (FEM), Rua Mendeleyev, 200, Cidade Universitária, 13083-860, Campinas, São Paulo, Brazil

ARTICLE INFO

Keywords:
Rock type classification
Core drill plug classification

ABSTRACT

We investigate machine-learning techniques for rock-type classification. A throughout literature review (considering the machine-learning technique, number of classes, rock types, and image types) presents a diversity of datasets employed and a wide range of classification results as well as multiple problem formulations. Throughout the discussion of the literature, we highlight some common machine-learning pitfalls and criticize the decisions taken by some authors on the problem formulation. We present an experimental contribution by evaluating the classification of seven types of rocks found in carbonate reservoirs along with state-of-the-art Convolutional Neural Networks (CNNs) architectures available through a well-known open-source library. For this experimentation, we detail the preparation of the dataset of drill core plugs (DCPs), the experimental setup itself, and the obtained results considering the normalized accuracy and the traditional accuracy as metrics. We performed the manual background segmentation of the employed dataset of DCPs; so the results reported are not influenced by the background of the images. We evaluate top-1, top-2, and top-3 performance for the problem. We apply fusion of multiple CNNs for richer classification decisions. We also contribute by presenting the manual classification — human labeling by looking at the image on the computer screen — of the same seven-class dataset, performed by six non-geologist volunteers. Finally, we present a conclusion for the results obtained with our experiments and share valuable advice for researchers applying machine learning to rock classification.

Contents

1. Introduction 2
2. Related work 2
3. Experiments 6
   3.1. Dataset 6
   3.2. Experimental setup 7
   3.3. Results 8
   3.4. Manual classification 9
4. Key conclusions and recommendations 10
   4.1. Difficulties of the classification problem 11
   4.2. Recommendations and future work 11
   CRediT authorship contribution statement 12

* Corresponding author.
E-mail address: pedrormjunior@gmail.com (P. Ribeiro Mendes Júnior).

https://doi.org/10.1016/j.cageo.2024.105730
Received 7 November 2023; Received in revised form 21 May 2024; Accepted 13 September 2024
Available online 18 October 2024

Declaration of competing interest 13
Acknowledgments 14
Data availability 16
References 16

1. Introduction

Rock-type classification is a crucial aspect of geology that involves identifying and grouping different rocks according to their physical and chemical characteristics. This categorization usually considers mineral content, texture, and formation process. The uses of rock-type classification are diverse and varied. In petroleum geology, for example, rock classification helps explore and extract oil and gas, as some rock types are better suited for reservoirs than others. However, rock-type classification faces several challenges. One of the main difficulties is the variability of rock characteristics within a single classification, which can result in ambiguity and inconsistency in the classification outcomes. Technological limitations in accurately analyzing and interpreting rock properties also pose significant hurdles. Additionally, the ever-changing nature of rocks due to geological processes such as metamorphism adds complexity to the classification efforts.

In this work, we critically survey the literature for rock classification problems considering works that deal with the classification of drill core plugs (DCPs), borehole wall images (BWIs), drill cutting images (DCIs), well logs (WLs), borehole electrical images (BEIs), thin-section images (TSIs), X-ray computed tomographies (xCTs), and natural rock images (NRIs). Except for a few works considered, which describe the rock characterization by an expert geologist, we specify the machine-learning technique employed, *e.g.*, Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs; Cortes and Vapnik, 1995), Random Forests (RFs; Breiman, 2001), *hand-crafted features*, among others. Throughout the discussion, we highlight the similarities among the works as well as some pitfalls that seemingly occurred in the definition of the experimental setup by some authors. We also present a detailed summary (see Table 1), which allows us to observe the wide range of datasets employed, different problem formulations, and divergence of results obtained among the works.

Unlike the machine-learning literature, the literature on rock classification does not include well-known benchmarks used among researchers, such as the well-known ImageNet (Russakovsky et al., 2015) dataset. This hinders the comparison among the works in the hopes to discover a more effective technique for the specific problem.

In this work, we also investigate the problem of automatic DCP classification comprising seven types of rocks from carbonate reservoirs: crystal shrub, spherulite, grainstone/rudstone, calcimudstone, igneous, conglomerate, and shale.[1] For the sake of experimentation, we have evaluated multiple network architectures available through the PyTorch framework (Paszke et al., 2019). Details of these networks are available in Section 3.2.

Classification of DCPs is usually performed by a specialist through costly laboratory analyses (Caja et al., 2019), often using a microscope for analyzing *thin-sections* of the drill cores (DCs) (Gomes et al., 2020). Our work seeks an automatic classification system of those DCPs through digital images acquired, for instance, with professional digital single-lens reflex (DSLR) cameras. Such classification method can enable more affordable and faster classification of DCPs as well as aid in the learning process of novice geologists. However, our main objective with the experiments performed in our work is to draw conclusions upon the related work to be described in Section 2.

The classification of DCPs is a complex task, as it suffers from numerous problems, *e.g.*, *intra-class dissimilarity*, *inter-class similarity*, as well as *label non-agreement* (Paullada et al., 2021).[2] Also, as the images are sometimes acquired by a professional DSLR camera, it is common to find regions in the images that lack focus on the DCP, thus disturbing characteristics that could possibly be extracted and employed otherwise.

In this work, in addition to analyzing the employment of multiple neural network architectures for the problem, we analyze their combination through a Fusion *Fully-Connected* Network (Fusion-FCN), which consists of a multilayer perceptron (Bishop, 2006) that is previously learned on top of the output of each employed network architecture *fine-tuned* on the same problem.

The remainder of this manuscript is organized as follows: In Section 2, we present our survey on rock classification and, at the end, we describe aspects of our experimental work in comparison to related work. In Section 3, we present our dataset, devised experimental setup, and the obtained results, including those for Fusion-FCN. In the same section, we present the *manual classification* performed on the same dataset by six volunteers. Finally, in Section 4, we draw our main conclusions and recommendations.

2. Related work

In this section, we present a review of the literature on the rock classification problem, considering not only the papers that handle it through DCs but those that classify natural rock images (NRIs). There are also works that deal with borehole wall images (BWIs) — which are taken from within the borehole itself without the need to extract DC samples — and works that deal with borehole electrical images (BEIs). Among those dealing with DCs, there are works that focus on drill core plugs (DCPs), with thin-section images (TSIs), with well logs (WLs), and with X-ray computed tomographies (xCTs).

Folk (1959) describes an intricate *manual classification*[3] of limestones, proposing its division into three main families. Interestingly, although the authors in 1959 did not deal with automatic classification of the limestones, they argued that the task is not as straightforward as "something that one can plunge into immediately with 'cook book' in one hand and calculating machine in the other" (Folk, 1959). No classification accuracy is presented in that work, but the authors clearly expressed the non-agreement among geologists for the same classification task, which is a base factor that affects the automatic classification by any machine-learning method. See Table 1 with a summary of the reviewed works.

Haralick et al. (1973) performed one of the earliest automatic classification of rock images. In their work, proposing an image texture descriptor a.k.a. Haralick texture features, the authors employed a dataset of photomicrographs of sandstones, obtaining an 89% accuracy on distinguishing among 5 subcategories of this rock type.

Lepistö et al. (2005) worked with NRIs and employed the k-NN classifier along with *Gabor filtering* (Wanderley and Fisher, 2001) on a 4-class classification problem. The authors also compared this texture filtering approach applied in red, green, blue (RGB) and hue, saturation,

[1] Notice we consider the grainstones and the rudstones into a single class for the machine-learning classification problem.

[2] A study on *label non-agreement* on segmentation of rock images is given by Andrä et al. (2013). Folk (1959) also mention such "controversies" in geology, for the matter of limestone classification.

[3] Here, we refer to *manual classification* as the process of describing the rock types in geology, so, in this sense, it is the process of defining the *ground-truth*. Not to be confused with the manual classification we present in Section 3.4, which consists of visually classifying the images of DCPs by a human annotator.

Table 1
Related work breakdown for DC classification. Image types are as follows: drill core plug (DCP); borehole wall image (BWI); drill cutting image (DCI); well log (WL); borehole electrical image (BEI); thin-section image (TSI); X-ray computed tomography (xCT); and natural rock image (NRI). "**(?)**" indicates the respective information could not be obtained from the paper or it is unclear. "—" indicates the respective information *does not apply*.

Reference	Classification method	Image type	# of classes	Claimed accuracy	Dataset
Folk (1959)	*Manual*[m1]	TSI	3[c1]	—	Private
Haralick et al. (1973)	**(?)**	**(?)**[i1]	5	89%	**(?)**
Lepistö et al. (2005)	*k*-Nearest Neighbors (*k*-NN) (Bishop, 2006)	NRI	4	Around 80.77%	**(?)**
Linek et al. (2007)	Bayes' rule (Bishop, 2006)	BWI[i2]	6	89.9–98%	By Ocean Drilling Program (ODP)[d1]
Thomas et al. (2011)	Fuzzy logic (Zadeh, 1965), *k*-NN	DC, WL	3+1[c2]	94.29%	**(?)**
Chatterjee (2012)	SVM	DCI	6	96.2%	Private[d2]
Sharif et al. (2015)	**(?)**	NRI	**(?)**[c3]	80.3%[a1]	Private
Al-Mudhafar (2017)	Gradient Boosting Machine (GBM) (Friedman, 2001), Probabilistic Neural Network (PNN)	WL	3	95.81%	
Bestagini et al. (2017)	GBM	WL	9	Up to 53%	
Cheng and Guo (2017)	CNN	TSI	3[c4]	Up to 98.5%	Private[d3]
Shu et al. (2017)	SVM	NRI	9	Up to 96.71%	Private[d4]
Budennyy et al. (2017)	Decision tree (Bishop, 2006), RF	TSI	2, 3	90 & 96.1%	**(?)**
Leal F. et al. (2018)	SVM, logic function	WL, BEI	3	98.8, 94.0, & 94.6%	
Dakhelpour-Ghoveifel et al. (2018)	*Manual*[m2]	WL	5	—	Private[d5]
Karimpouli and Tahmasebi (2019)	SegNet (Badrinarayanan et al., 2017) (CNN)	TSI[i3]	5	96%	By Andrä et al. (2013)
Pascual et al. (2019)	CNN[m3]	NRI	2[c5]	99.6% & 89.3%	By Shu et al. (2017)
Ran et al. (2019)	CNN[m4]	NRI	6	97.96%[a2]	**(?)**
Caja et al. (2019)	SVM	TSI	4	**(?)**[a3]	Private[d6]
Hébert et al. (2020)	CNN[m5]	xCT	2, 4	100%[a2]	Private[d7]
Baraboshkin et al. (2020)	CNN	DCP	3	Up to 95%	
Su et al. (2020)	CNN[m6]	TSI	13	89.97%	Private[d8]
Alzubaidi et al. (2021)	CNN	DCP	3	93.12%	By Geological Survey of South Australia (GSSA)
Guojian and Peisong (2021)	ResNet (He et al., 2016) (CNN)	TSI	5[c6]	Up to 91.63%	Private[d3]
Almisned and Alqahtani (2021)	*Manual*[m7]	—	—	—	Private[d9]
Günther et al. (2021)	**(?)**	DC	**(?)**	Around 80% (segmentation)	
Fu et al. (2022)	CNN	DCP	10	99.60%	By China Geological Sample Information (CGSI)
Zheng et al. (2024)	ResNet (CNN)	TSI	6	99%	Private
Our work	CNNs[m8], Fusion-FCN[m9]	DCP	7	≈69%[a4]	Private[d10]

[m1] The authors present a detailed study on the classification of limestones.
[m2] The authors present a study on *rock typing* in *transition zones* based on *irreducible water saturation*.
[m3] The authors designed a 3-layer CNN.
[m4] The authors propose an architecture named Rock Types deep CNN (RTCNN).
[m5] The authors simply employ a pre-trained Inception V3 (Szegedy et al., 2016).
[m6] The authors propose an architecture named Concatenated Convolutional Neural Network (Con-CNN).
[m7] The authors present a detailed study on the classification of *soft rocks*.
[m8] We perform experiments with the networks listed in Section 3.2.
[m9] We describe Fusion-FCN in Section 3.2.
[i1] The authors reported using photomicrographs of sandstones.
[i2] Acquired with Formation MicroScanner (FMS) tool.
[i3] The authors mention the use of scanning electron microscopy (SEM).
[c1] The authors present three main limestone families.
[c2] The authors consider the *no-core* as a fourth class.
[c3] The authors perform analysis of parametrization for the dataset according to the number of images; see discussion in Section 2.
[c4] The authors perform the classification of feldspar sandstones images as having *coarse*, *medium*, and *fine* granularity.
[c5] The authors consider a binary classification problem consisting of breccia vs. non-breccia.
[c6] The authors classify feldspar sandstones into five types of granularity: *fine-grained*, *fine-medium-grained*, *medium-grained*, *medium-coarse-grained*, and *coarse-grained*.
[a1] According to an arbitrary *classification accuracy score*.
[a2] See discussion in Section 2.
[a3] The authors simply compare the percentage of samples classified to each class with the expected percentage of rock types on the reservoirs.
[a4] Our experimental setup is described in Section 3.2 and results are reported in Section 3.3.
[d1] Data from "hole 1203A" (Division of Marine Large Programs, 2023) drilled at Detroit Seamount.
[d2] Limestones from a mine in western India.
[d3] Feldspar sandstones from Ordos, China.
[d4] From Department of Earth Sciences in Western University.
[d5] Iranian carbonate reservoirs.
[d6] Sixteen cutting samples from two reservoirs.
[d7] Carbonates from *Estaillades* and *Savonnieres* & sandstones from *Fontainebleau* and *Berea*.
[d8] Online data (Gill, 2023; Ward's Science, 2023) and data from Zhejiang University.
[d9] *Soft rock* samples from Al-Kharj, Riyadh Province of central Saudi Arabia.
[d10] Our dataset is described in Section 3.1.

intensity (HSI) color spaces, concluding that in RGB space results are clearly better. The dataset preparation for the classification experiments is mentioned as done by "an expert [...] based on [image] color and texture properties", however, the four classes considered for the classification problem are not named and it is not clear if each class refers to a specific rock type. A *leave-one-out cross-validation* (Bishop, 2006) protocol was employed for reaching the accuracy of approximately 80.77%.

Linek et al. (2007) employed BWIs (Deng et al., 2018) for estimating lithofacies. Those images are usually taken throughout the borehole itself, *i.e.*, for acquiring the images, no extraction of rock samples is necessary. The data employed were acquired with the FMS (Ekstrom et al., 1986; Chen et al., 1987; Badr and Ayoub, 1989) sonde; to describe the data, the authors employed Haralick texture features (Haralick et al., 1973; Haralick, 1979) and wavelet transforms (Wouwer et al., 1999). The class division performed by the authors consists of three classes within volcaniclastic rocks (breccia, layered, and resedimented) and three subtypes of igneous rocks (massive, pillow, and vesicular).

The accuracy reported by the authors for the 6-class problem ranges from 89.9% to 98% for data from "hole 1203A" (Division of Marine Large Programs, 2023) at the Detroit Seamount obtained through the ODP. This accuracy, however, seems to be overestimated; the authors report using a portion of the test data (*i.e.*, the entire BWI) for training the classifier.

Thomas et al. (2011) classified three rock types (sand, shale, and carbonate cements) and considered a fourth *no-core* class to refer to the parts of the images mainly regarding the background. By employing fuzzy logic and k-NN, the authors classified segmented regions of the DC photographs, but the reported accuracy was based on just 315 "objects" classified and, among the 4 classes, it was of 94.29%.

The description presented by Thomas et al. (2011) can raise a number of doubts regarding the correctness of the experimental setup employed, or at least it might not be comparable to our work on the same experimental classification scenario. Firstly, the authors mention an interactive process of adjusting the classification method to the data[4] until obtaining the desired accuracy.[5] Another alternative presented by the author is to *fine-tune* the interactive classifier to new data when working on a new dataset. Secondly, the *ground-truth* employed for the experiments is assumed to be one defined by *visual inspection* of a geologist, which means that any visual deception that can misguide the classification system could also misguide the geologist. Our experiments on the manual classification presented in Section 3.4 show this fact. The high classification accuracy reported by the author might also come from this inherent "information alignment" between the *ground-truth* and appearance of the DCs.

Chatterjee (2012) employs hand-crafted feature extractors along with a multiclass-from-binary extension — based on the one-vs-all (Rocha and Goldenstein, 2014) approach — of the well-known SVM classifier for a 6-class classification problem of drill cutting images (DCIs), with images segmented based on watershed (Beucher, 1979) algorithm. The SVM, hyper-parameterized with genetic algorithm (Katoch et al., 2020), was applied on top of 189 pre-extracted features comprising color (112 features), morphology (28 features), and texture (49 features). The reported accuracy reaches 96.2% in their experiments, based on upper gray limestone, clay, dark gray limestone, pink limestone, greenish gray limestone, and weathered limestone, with 20 samples obtained per class. The authors also compared the SVM with a neural network with one hidden layer whose input is the set of 189 pre-selected features, evincing the superiority of the SVM in this case.

Sharif et al. (2015) also employed Haralick texture features, 13 of them, for the classification igneous, sedimentary, and metamorphic rocks. The authors properly reported the DSLR camera configuration (*e.g.*, shutter speed, aperture) for acquiring their employed NRI dataset. They also analyzed which of the Haralick texture features were more important for the problem.

The experimental setup devised by Sharif et al. (2015) was noticeably unconventional: they elaborated a *classification matrix* that differs from a confusion matrix (Bishop, 2006)[6] in which the rows and columns represented the samples employed for analysis. From those *classification matrices*, they calculated an arbitrary *classification accuracy score* (Sharif et al., 2015, Section 4), and the reported "accuracy" reached 80.3%.

Usually, rock properties are extracted from DCs or DCPs from lab experimentation to obtain the WLs. Anyhow, the methods employed by Al-Mudhafar (2017) work in the opposite direction, *i.e.*, obtaining the lithology from the WLs. The authors employ GBM and PNN (Mao et al., 2000) for the 3-class problem and present the accuracy of 95.81%.

Bestagini et al. (2017) also dealt with rock-type classification through WLs by employing GBM, but they reported a classification accuracy of up to 53% for the 9-class problem.[7] They considered nonmarine sandstone, nonmarine coarse siltstone, nonmarine fine siltstone, marine siltstone and shale (as a single class), mudstone, wackestone, dolomite, packstone-grainstone, and phylloid-algal bafflestone as classes.

Cheng and Guo (2017) performed the classification of three granularities of feldspar sandstones from an oil field in Ordos, China. They demonstrated an accuracy of 98.5% in that classification problem by experimenting with multiple CNN architectures, each with no more than six layers. For the task, the employed data consist of TSIs.

Shu et al. (2017) leveraged a feature representation method (Coates and Ng, 2012) based on k-means (Bishop, 2006) along with SVM classifier. The authors also investigated manually selected statistical features and the employment of self-taught learning (Raina et al., 2007) as an alternative feature representation. The problem consists of 9 classes of images provided by the Department of Earth Sciences at Western University, which are limestone, volcanic breccia, oolitic limestone, dolostone, rhyolite, granite, andesite, peridotite, and red granite. The accuracy obtained with feature representation through k-means reaches 96.71%, while just 90.32% is obtained with self-taught learning. They showed that, while self-taught learning appears to be a promising technique, it leads to lower accuracy than that obtained through manual statistical features (up to 96.24%).

Budennyy et al. (2017) employed TSIs to classify among three rock types: sandstone, limestone, and dolomite. They first performed segmentation of regions of interest in the image by employing the watershed method. After the *description* of each region, decision tree was used for classification among the rock types. The authors also performed the classification of the mineral composition of sandstones into greywacke or arkose, through the RF classifier.

Leal F. et al. (2018) considered binary classification for rock sections also basing the decision on WLs besides BEIs. They employed two SVM models; one to decide if the rock interval is positive for fossiliferous limestone and another to predict the presence of calcareous shale interbedded with limestone. As for deciding if an interval of the DCP contains *laminated calcareous rocks*, both SVM models should predict if the neutron/photoelectric factor of the section should be higher than or equal to 4. The authors reported individual accuracy for those three binary classifiers.

[4] "[...] Wrongly classified objects, if any, are moved to the correct class either by adding or removing a few sample objects to the training sample sets, to attain the desired classification. [...]" — Thomas et al. (2011), second paragraph, p. 106.

[5] "Run the protocol iteratively while editing the class hierarchy mask to correct misclassifications on each iteration to suit the new field"., Thomas et al. (2011), p. 106.

[6] Each row of their *classification matrices* (Sharif et al., 2015, Tables 2, 3, and 5) contains numbers in the range of $[0, 1]$ and sums up to more than 1.0.

[7] This reported accuracy is more in tune with the results of the experiments we prepared for this manuscript, although we have obtained a slightly higher accuracy for a problem with fewer classes.

Dakhelpour-Ghoveifel et al. (2018) presented a study on *rock typing* based on *irreducible water saturation*. Unlike a traditional classification problem, they presented how to estimate the *water saturation map* for correct rock typing for the special case of *transition zones* of the reservoir. The authors suggested their data come from Iranian carbonate reservoirs, but they reported no classification accuracy among the five rock types they consider in their work.

Karimpouli and Tahmasebi (2019) employed the encoder–decoder SegNet architecture for segmentation of Berea sandstone into *pore space*, quartz, K-feldspar, zircon, and *other minerals* (*e.g.*, clays). Due to the lack of data faced by the authors, they used *data synthesis*/data augmentation (Shorten and Khoshgoftaar, 2019) for training the network. The reported accuracy is of 96% for this 5-class segmentation task.

Pascual et al. (2019) worked with NRIs and, compared to the previous work of Shu et al. (2017), their main novelty is the use of data augmentation for training a CNN. They considered a 2-class problem consisting of breccia vs. non-breccia classification. By employing 85% of the dataset for training a 3-layer CNN, the reported accuracy is of 99.6%. As for a 5-layer CNN, accuracy was of 89.3%. The authors also concluded that the 3-layer CNN outperformed the SVM for the task. They also reported that the employed dataset was obtained by Shu et al. (2017).

Ran et al. (2019) also tackled the problem of NRI classification, and considered a 6-class problem consisting of mylonite, granite, conglomerate, sandstone, shale, and limestone. The authors claimed that the few-layered CNN named by them as RTCNN obtained improved accuracy of 97.96% when compared to SVM, AlexNet (Krizhevsky et al., 2012), VGG (Simonyan and Zisserman, 2015), and Inception V3 alternatives, despite its simpler structure. However, the difference between their proposed network and the Inception V3 is of just 0.86% in classification accuracy.

A major technical concern that arises from the work of Ran et al. (2019) concerns the correctness of the dataset split into *training*, *validation*, and *test* sets, as those sets are split after the extraction of the 24 315 patches from the 2290 field photographs comprising the original dataset. The authors do not clearly state that patches from the same original image should fall into the same *training*, *validation*, or *test* partitions. By not ensuring it, the network could be tested with data similar to those used for training (*i.e.*, patches from the same original image), which would result in data contamination (Magar and Schwartz, 2022) if this is indeed the case for their experimental setup.

Caja et al. (2019) simply applied an SVM classifier on *regions of interest* to classify them among four classes: quartzites, siltstones, claystones, and carbonates. They employed 16 cutting samples from two reservoirs, from which they extracted TSIs for defining the classification problem.

Hébert et al. (2020) employed 3D xCT samples from four reservoirs that comprise carbonate and sandstone classes of rocks. The authors also defined a classification problem consisting of recognizing the 4 rock formations their data come from, *i.e.*, *Estaillades*, *Savonnieres*, *Fontainebleau*, and *Berea*. They followed an approach similar to what we are employing for the experiments in this manuscript, which consists of *fine-tuning* a network pre-trained on ImageNet dataset, which is a large benchmark employed by *computer vision* community consisting of a thousand classes of general objects. In the case of Hébert et al. (2020), the employed network architecture was the Inception V3. The authors also worked on predicting porosity from the 3D samples, employing a diverse network as suggested by Sudakov et al. (2019).

The authors reported 100% accuracy for both the 2- and 4-class classification problems. The main reason for such a high accuracy might be a dataset bias: (1) For the carbonate vs. sandstone classification problem, it is worth stating that samples of carbonates come just from *Estaillades* and *Savonnieres* while all their examples of sandstone come from the two other formations: *Fontainebleau* and *Berea*. (2) For the 4-class problem, the bias is more evident since the objective is to classify the rock formations themselves: a complex model like the Inception V3 is well-known to easily learn the data acquisition setup more than the content of the very image.[8]

Baraboshkin et al. (2020) adopted well-known network architectures such as AlexNet, VGG, GoogLeNet (Szegedy et al., 2015), and ResNet for a 3-class problem consisting of sandstone, limestone, and shale lithologies and they claim to have obtained an accuracy of up to 95%. For obtaining this accuracy, the authors acquired 2000 images of boxes of DCs, then cropped 20 000 images of 10×10 cm from them, establishing the dataset. The authors do not clarify how the split into *training*, *validation*, and *test* is performed, therefore, the high accuracy can be due to a kind of data contamination as it can happen that a 10×10 cm image can fall into training set while a neighbor 10×10 cm image appears in test. Based on their data, a reasonable criteria to apply for establishing the experimental setup is to ensure that all 10×10 cm images from a single box of cores fall into a single part of the split (*i.e.*, *training*, *validation*, or *test*).

Su et al. (2020) worked with TSIs extracted with plane polarized light (PPL) and crossed polarized light (XPL) techniques. Comprising borrowed data (Gill, 2023; Ward's Science, 2023) and data from their university, the employed dataset includes 13 rock types: andesite, granite, peridotite, gabbro, rhyolite, tuff, diorite, phonolite, basalt, syenite, limestone, sandstone, and schist. The authors employed a straightforward multilayer CNN as *backbones* to their network approach named Con-CNN, which consists of employing the backbones for classifying the PPL, XPL, and a *comprehensive image* (containing information from both PPL and XPL) independently, then having a combination of the predictions as the final prediction. The reported accuracy is of 89.97%. They also experiment with LeNet (LeCun et al., 1998) and VGG as well as the ResNet architectures as the backbone, but reported worse results with those.

Alzubaidi et al. (2021) also employed well-known network architectures such as ResNeXt (Xie et al., 2017), ResNet, and Inception V3 for discriminating among sandstone, limestone, and shale. The claimed accuracy in their case is of 93.12%. Their employed data comes from the GSSA. From a total of 858 images comprising the three classes, they employed a patch-based approach for classification. The authors did not clarify how they split *training*, *validation*, and *test* sets whether by images (the correct way) or by patches (which would lead to data contamination).

Guojian and Peisong (2021) simply employed the ResNet architecture for the rock classification problem based on TSIs. Instead of classifying rock types, the authors classified feldspar sandstones from Ordos, China,[9] according to 5 types of grain sizes. The maximum reported accuracy for this problem is 93.12%.

Almisned and Alqahtani (2021) dealt with manual classification of *soft rocks*, *i.e.*, rocks with a strength between *soils* and *hard rocks* that do not crumble when immersed in water (Almisned and Alqahtani, 2021). As the authors described a manual intricate classification of *soft rocks*, they did not present automatic classification results for the samples obtained in Al-Kharj, Riyadh Province of central Saudi Arabia.

Günther et al. (2021) presented an overview of the process for automatizing the classification of DCs from its acquisition, considering the intermediate steps that require automation, but the authors only performed experiments for the segmentation of the box of DCs on the image and even no details are given in regard to the segmentation method or the possible classification method.

Fu et al. (2022) worked on a dataset of DCPs made available by CGSI considering a 10-class problem (diabase, diorite, gneiss, granite, limestone, marble, monzonite, mudstone, shale, and siltstone). Like

[8] One should note that we are not stating that it was the only factor of success in the classification performed by Hébert et al. (2020); as we can see in Table 1 of their work, the image content of their examples can also be visually distinguishable per rock formation.

[9] Possibly the same dataset employed by Cheng and Guo (2017).

other works on the same problem, the authors employed some of the network architecture previously published in the literature, in this case, the ResNeSt (Zhang et al., 2022) *backbone*, comparing it with a few others (DenseNet (Huang et al., 2017), ResNet, and VGG); we do not consider ResNeSt for the experiments performed in this manuscript. The problem dealt with by the authors seems to have more well-separated classes and the reported test accuracy is of 99.60%. They established a balanced problem ensuring 1500 images of 256 × 256 pixels from each class. These images were cropped from original examples of CGSI dataset. As multiple images can be cropped along the same original image of a DC, a correct approach for a machine-learning experiment would be to ensure that images cropped form the same DC image to appear on the sample part of the data split (*training*, *validation*, or *test*), however, the authors do not mention if this restriction was taken into account.

A primary challenge in rock-type classification is ensuring the explainability of machine-learning models and enhancing their interpretability to clarify the rationale behind their predictions. Zheng et al. (2024) have tackled this concern by proposing and developing an interpretable rock classification deep-learning model that integrates geological knowledge. Specifically, their focus is on sedimentary rock classification (quartz arenite, feldspathic arenite, lithic arenite, siltstone, oolitic packstone, and dolomite) based on TSIs, emphasizing the critical role of interpretability alongside geological expertise. Their research not only underscores the significance of interpretability but also introduces the attention-based dual-modal SedNet model that achieves high accuracy (99%) while enabling interpretable feature extractions. This work employed 1356 cross-polarized light photomicrographs acquired from 15 examples covering the six employed classes, however, the authors do not make it clear how the split of the images were done for the experimental setup. No guarantee is given for ensuring two images extracted close by each other in the sample example fall into the same part of the split.

In our work, we investigate multiple network architectures (Section 3.2) for DCP classification. Furthermore, we investigate *intermediate fusion* (Boulahia et al., 2021) through a Fusion-FCN (see Section 3.2). Our dataset consists of seven classes of rocks (crystal shrub, spherulite, grainstone/rudstone, calcimudstone, igneous, conglomerate, and shale), obtained through *Shell Brazil*, for which we will later present an overview (Section 3.1). In our work, we show that, for a 7-class classification problem of carbonate rocks, accuracy can slightly exceed the 69% margin (Section 3.3). Furthermore, we confirm the obtained accuracy is reasonable by comparing it with the manual classification of the dataset performed by six volunteers.

Previous work dealing with DCP present high accuracy (≈95%), however, some of them does not clarify the experimental setup employed, *i.e.*, if it was correctly defined. For instance, Baraboshkin et al. (2020) do not clarify if the 10 × 10 cm images of the DCs are ensured to be on the same part of the *training*, *validation*, and *test* split. Similarly, Alzubaidi et al. (2021) and Fu et al. (2022) do not mention if the split is correctly performed based on images or based on patches, which would lead to data contamination.

Most of the works we have considered present a high accuracy greater than 90%. The reason might be due to the data contamination problem we have discussed or some of these works are dealing with more separable types of rocks. Anyhow, the only works that present accuracy lower than 90% are the following. Haralick et al. (1973) obtains 89% of accuracy in a 5-class problem, however, they did not specify the classification method nor the image type. Lepistö et al. (2005) presented around 80.77% accuracy for a 4-class problem with the well-known k-NN classifier for NRIs; although employing the *leave-one-out cross-validation* protocol, their experimental setup does not seem to lead to the common data contamination previously discussed as they do not extract cropped images from the originally acquired images. Sharif et al. (2015), on the other hand, also working with NRIs, report 80.3% accuracy, however, their description of the experimental setup does not make it clear no data contamination is present. Su et al. (2020) report 89.97% accuracy, even though their experimental setup seems to suffer from data contamination as they treat the 108 224 × 224 images extracted from a 2688 × 2016 image as independent. And Bestagini et al. (2017) report 53% accuracy in the more challenging scenario of classifying the rock types from WLs.

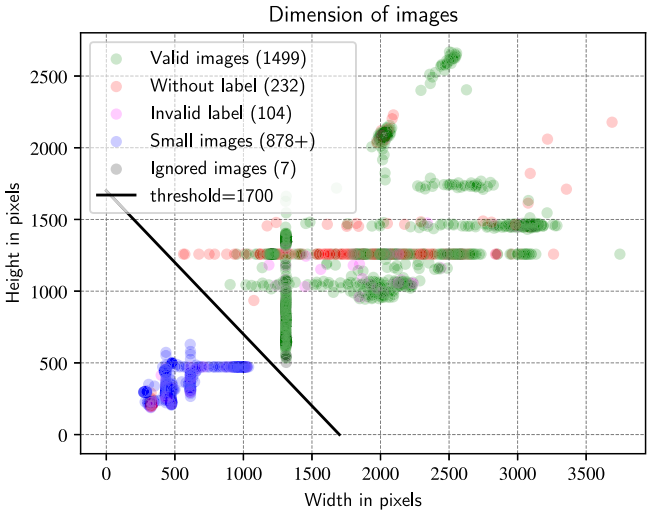


Fig. 1. Overview of the dataset by the size of the images. Each dot represents one image from the original dataset. Green ones represent the images from the seven classes we are employing in this work. Red ones are the examples without annotated labels. Magenta ones are from the classes not considered in this work. *Too small* images (below the threshold) are marked in blue in case not already marked in red or magenta. The threshold of 1700 is applied to the sum of the height and width of each image to be considered valid or *too small*. Finally, the few ignored images marked in black are those from which we could not extract patches (more details in Section 3.4).

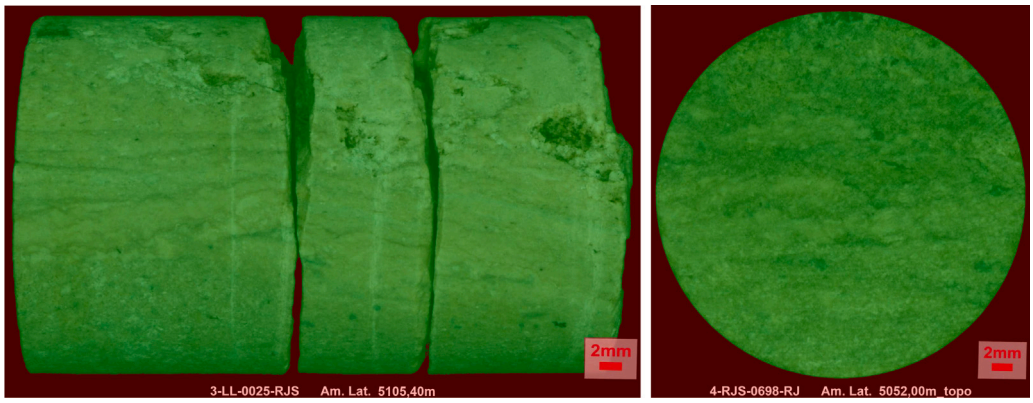
3. Experiments

In this section, we describe the dataset adopted for our experiments (Section 3.1), the experimental setup (Section 3.2) along with Fusion-FCN definition, and the results (Section 3.3) obtained for the rock classification problem. Herein, we also present (Section 3.4) the results obtained for the manual classification of the dataset described in Section 3.1 by six volunteers from our research lab.

3.1. Dataset

The dataset employed in this work was kindly provided by *Shell Brazil*, comprising images obtained from wells in pre-salt carbonate fields in Brazil. Originally, it comprised 14 classes of rock types from which we selected the seven most populous ones for experimentation. All images that contained no label information were ignored. The original dataset also had *small images* of DCPs, which seem to be resampled from higher resolution images; those were also ignored by the criterion that the sum of the height and the width of the image should be greater than 1700 pixels. The final number of images employed for the experiments is 1499. These employed images comprise 14 wells. In Fig. 1, we present an overview of the dataset in terms of the dimensionality of the images in it.

Those 1499 images employed for experiments were then manually segmented for the region containing the DCP. Two examples of segmented DCPs are shown in Fig. 2. For the experiments in our work, as segmentation is employed, we ensure that the 224 × 224 patches extracted from the images entirely fall within the masked region for the DCP. In Fig. 3, we present examples of patches extracted per class.



(a) Long view of a DCP. (b) Top view of a DCP.

Fig. 2. Examples of DCP manual segmentation. Greenish regions are the regions marked as referring to the DCP and reddish regions are those associated with the background.

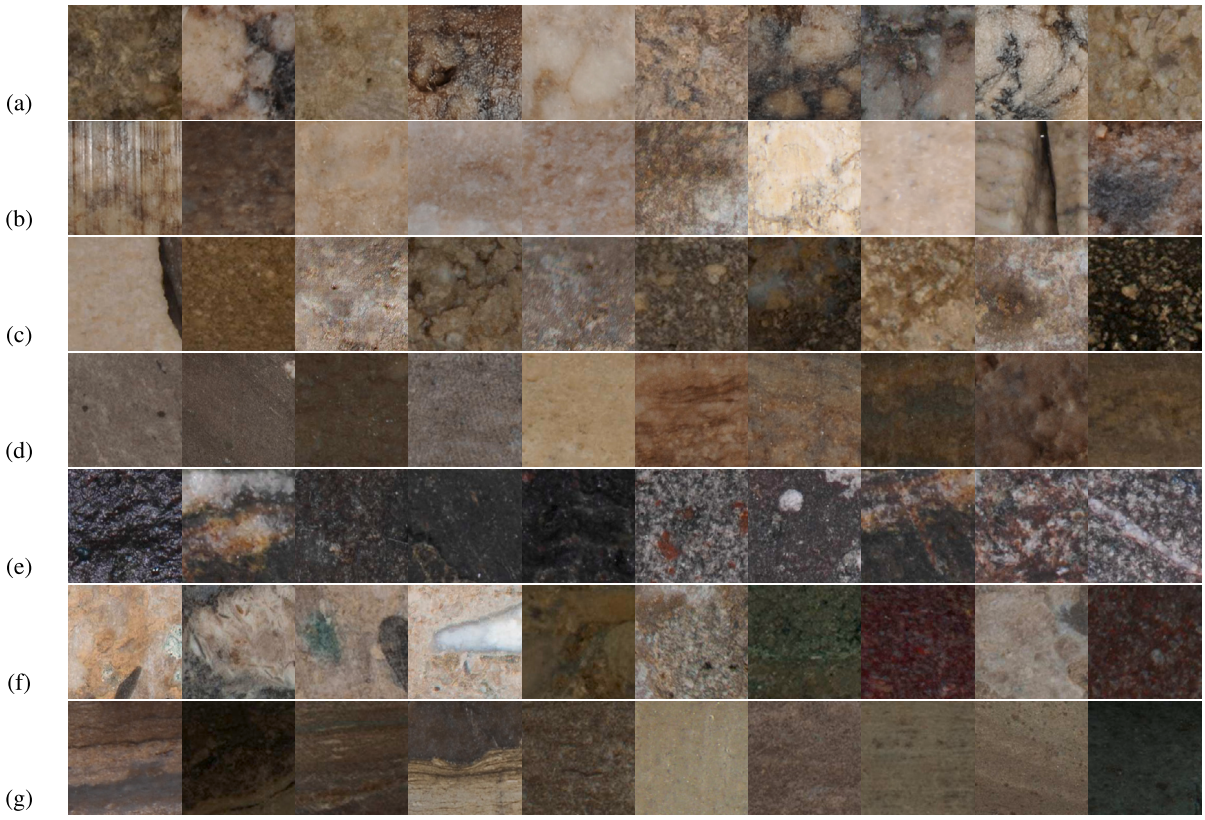


Fig. 3. Examples of patches extracted per class. For each class, 10 random patches were selected. Classes are as follows: (a) Crystal shrubs; (b) Spherulites; (c) Grainstones/Rudstones; (d) Calcimudstones; (e) Igneous; (f) Conglomerates; (g) Shales.

3.2. Experimental setup

Among the 1499 valid images mentioned in Section 3.1, we decided to split them into multiple *training*, *validation*, and *test* hold-out partitions to better estimate the obtained accuracy for the problem. Eight of those partitions were defined for performing experiments and reporting normalized accuracy (NA) and accuracy (ACC). Each partition contains roughly 60%, 16%, and 24% of the DCPs in *training*, *validation*, and *test* sets, respectively. A few DCPs contain two associated images, one with *long view* and the other with *top view*, as exemplified in Fig. 2. For those cases, for each of the eight partitions, we ensure both images fall into the same set, *i.e.*, *training*, *validation*, or *test*, so as to avoid data contamination. In Table 2, we show details of each of the eight partitions.

Throughout those experiments, we employ a *patch-based approach*, *i.e.*, we extract multiple 224 × 224-wide patches from each image and use them as examples (from the point of view of the trained networks). In this vein, we are able to evaluate accuracy for the *image-wise classification* as well as for the *patch-wise classification*. The former is possible by employing a straightforward *voting scheme*: each test image is classified as belonging to the class more often predicted by its patches.

For extracting patches from *training images*, we evaluate two possible approaches,

(1) employing pre-extracted patches in a *stride-based manner* and
(2) *extracting patches on the fly* by ensuring each one falls within the masked region for the DCP.

Table 2
Overview of the partitions of the dataset for experiments. For each of the 8 partitions (**Pt**), we have the number of DCPs (**#Cor**) and the number of images (**#Img**) associated with each part (**Pa**) — *training* (**Tr**), *test* (**Te**), and *validation* (**Va**) — and associated with each class (**C1**, **C2**, **C3**, **C4**, **C5**, **C6**, and **C7**). For each partition and for each part, we also show the minimum number among the classes (**Min**) and the sum of quantities among the classes (**Total**).

	Pa	Pt	C1	C2	C3	C4	C5	C6	C7	Min	Total
#Cor	Tr	1	156	134	115	64	33	22	19	19	543
	Te		55	56	43	30	20	9	4	4	217
	Va		38	43	25	19	10	4	6	4	145
#Img	Tr		261	209	195	112	49	41	28	28	895
	Te		91	90	74	53	33	17	7	7	365
	Va		65	68	42	31	17	7	9	7	239
#Cor	Tr	2	160	135	106	76	29	19	18	18	543
	Te		56	62	41	22	19	9	8	8	217
	Va		33	36	36	15	15	7	3	3	145
#Img	Tr		266	214	174	134	45	35	27	27	895
	Te		95	96	75	38	32	17	12	12	365
	Va		56	57	62	24	22	13	5	5	239
#Cor	Tr	3	151	135	106	69	41	22	19	19	543
	Te		64	59	42	25	13	8	6	6	217
	Va		34	39	35	19	9	5	4	4	145
#Img	Tr		258	215	185	120	65	40	25	25	908
	Te		106	91	68	43	20	16	11	11	355
	Va		53	61	58	33	14	9	8	8	236
#Cor	Tr	4	156	144	104	67	36	22	14	14	543
	Te		57	53	54	24	13	7	9	7	217
	Va		36	36	25	22	14	6	6	6	145
#Img	Tr		260	234	182	119	57	40	19	19	911
	Te		93	80	87	36	20	13	15	13	344
	Va		64	53	42	41	22	12	10	10	244
#Cor	Tr	5	151	134	109	62	51	21	15	15	543
	Te		58	62	46	28	7	8	8	7	217
	Va		40	37	28	23	5	6	6	5	145
#Img	Tr		247	210	187	106	80	40	24	24	894
	Te		98	104	77	49	11	15	11	11	365
	Va		72	53	47	41	8	10	9	8	240
#Cor	Tr	6	153	137	104	71	38	20	20	20	543
	Te		54	63	42	28	17	7	6	6	217
	Va		42	33	37	14	8	8	3	3	145
#Img	Tr		258	223	176	121	60	36	31	31	905
	Te		87	96	69	51	28	14	7	7	352
	Va		72	48	66	24	11	15	6	6	242
#Cor	Tr	7	150	134	110	76	35	17	21	17	543
	Te		57	57	47	26	17	8	5	5	217
	Va		42	42	26	11	11	10	3	3	145
#Img	Tr		247	215	182	132	55	30	30	30	891
	Te		100	87	85	42	24	15	8	8	361
	Va		70	65	44	22	20	20	6	6	247
#Cor	Tr	8	151	137	116	61	40	22	16	16	543
	Te		65	56	38	32	12	7	7	7	217
	Va		33	40	29	20	11	6	6	6	145
#Img	Tr		255	214	202	106	63	39	25	25	904
	Te		104	91	61	55	18	14	12	12	355
	Va		58	62	48	35	18	12	7	7	240

Even when approach (2) is employed, both *validation* and *test* sets still have their examples always extracted in a *stride-based manner* with *stride* equal to 100 pixels.[10] We observed a clear advantage of approach (2) compared to approach (1) for training, possibly due to the higher variability of input data given to the networks when that approach is employed, so the results we report in Section 3.3 follow approach (2).

[10] The number of patches (examples) extracted per image by the stride-based patch extraction depends on the area of the masked DCP. For stride equal to 100 pixels, the three images in our dataset with fewer patches extracted had 3, 4, and 8 patches; the three images with the highest number of patches extracted had 390, 390, and 392. The average number of patches extracted for the dataset images is 133.12 and the median is 87.

For each partition, we performed experiments with the networks listed in Table 3 and report their results individually. Based on some initial experimentation with InceptionV3,[11] we fixed the employed optimizer to be the Stochastic Gradient Descent (SGD) (Sutskever et al., 2013) with `lr`=1×10^{-4} (except for Fusion-FCN; see next paragraph) and `momentum`=0.9. Every model was trained with *cross-entropy* loss and training stopped when loss value had not improved for 100 sequential epochs, a.k.a., *early stopping* with *tolerance* of 100. We employed PyTorch's implementation (Paszke et al., 2019) for all networks starting from the best available pre-trained weights associated with each architecture, as specified in Table 3 itself.

In Table 3, we also described the Fusion-FCN that we employ for *fusing* the output of every other network listed in Table 3 and learning two fully-connected layers (a.k.a. multilayer perceptron) on top: 30 *units* first, then 7 units as output, referring to the number of classes. One should observe that the number of inputs for the learned fully-connected layers is 126 features by receiving the output of the 18 networks, containing 7 units each. For Fusion-FCN, we observed the need to employ a lower *learning rate*, *i.e.*, `lr`=1×10^{-6}. Later, in Section 3.3, we also present results for this *fusion* technique.

Furthermore, we evaluate a few traditional classification methods applied on top of the features extracted along with the networks specified in Table 3. In our case, we consider k-NN, RF, and SVM as well as straight Softmax classification performed by each network.

In addition to evaluating network efficacy for the *top*-1 classification of rock images into the seven previously specified classes, we also assess the performance for *top*-2 and *top*-3 *classification*, *i.e.*, NA and ACC when the correct class appears among the *top*-2 and *top*-3 classes, respectively, returned by the classification method as most likely ones. For instance, when evaluating *top*-3 classification, we consider that the classification method correctly classified an example when the correct class is among the 3 most probable ones estimated by the classifier. In practical scenarios, we devise that a user of the system, with limited lithographical knowledge, knowing the limitation of the classification accuracy, could consider with their expertise the *top*-2 or *top*-3 results.

Complementing those analyses, we present the manual classification performed by six members of our research team later in Section 3.4. In this case, we do not employ eight partitions as before; instead, we establish a single partition maintaining the same percentage of DCPs for *training*, *validation*, and *test* sets, *i.e.*, roughly 60%, 16%, and 24%, respectively. Each participant of the manual classification could indistinguishably use both *training* and *validation* sets for individual "training". That is, while classifying each of the images in the *test* set, each participant has *training* and *validation* images available for consultation along with the corresponding *ground-truth*. Each participant classified each *test* image as belonging to one of the 7 classes, *i.e.*, we just consider *top*-1 classification in this analysis. None of the 6 participants is a geologist, so we expect this experiment to solely expose the ability of a human to identify the most prominent visual features that are distinguishable among the considered classes. Further details regarding the *experimental setup* for the manual classification are presented in Section 3.4, along with the obtained results.

3.3. Results

For each network listed in Table 3, we present their results in Table 4. In this table, we show results for straight network classification in the Softmax column as well as for the k-NN, RF, and SVM classifiers. These three classifiers were run based on the 7-dimensional features output from the respective network with patch-based training examples

[11] We employed InceptionV3 initially for experimentation based on 299 × 299-wide patches. By establishing 224 × 224-wide patches, whose size is appropriate for the remaining networks, NA for InceptionV3 degraded by around 25%, then we decided to exclude this network for the overall analysis.

Table 3
Networks employed for experimentation. The networks are sorted based on the **# of parameters**. **Type** is named according to PyTorch's implementation (Paszke et al., 2019) as well as **Weights**, in which **w1**, **w2**, and **w3** refer to `IMAGENET1K_V1`, `IMAGENET1K_V2`, and `IMAGENET1K_SWAG_E2E_V1`, respectively. The Fusion-FCN fuses the output of the other listed networks (after they are *fine-tuned* to the problem) and learns two fully-connected layers on top.

Architecture	Type	Weights	# of parameters	# of layers
SqueezeNet (Iandola et al., 2016)	1_1	**w1**	726 087	52
MobileNet V2 (Sandler et al., 2018)	–	**w2**	2 232 839	158
MobileNet V3 (Howard et al., 2019)	Large	**w2**	4 210 999	174
MNASNet (Tan et al., 2019)	1_3	**w1**	5 010 223	158
ShuffleNet V2 (Ma et al., 2018)	x2_0	**w1**	5 359 339	170
GoogLeNet	–	**w1**	5 607 079	173
DenseNet	161	**w1**	26 487 463	484
AlexNet	–	**w1**	57 032 519	16
ResNet	152	**w2**	58 158 151	467
EfficientNet (Tan and Le, 2019)	B7	**w1**	63 804 887	711
ResNeXt	64X4D	**w1**	81 420 615	314
SwinTransformer (Liu et al., 2021)	B	**w1**	86 750 399	329
EfficientNet V2 (Tan and Le, 2021)	L	**w1**	117 243 239	897
Wide ResNet (Zagoruyko and Komodakis, 2017)	101_2	**w2**	124 852 039	314
VGG	BN	**w1**	139 609 927	70
ConvNeXt (Liu et al., 2022)	Large	**w1**	196 241 095	344
VisionTransformer (Dosovitskiy et al., 2021)	H_14	**w3**	632 198 407	392
RegNet (Radosavovic et al., 2020)	128GF	**w3**	637 471 645	368
Fusion-FCN		Fine-tuned	2 242 996 339	5594

extracted from training images with the previously mentioned stride (Section 3.2).

The results for *top*-2 and *top*-3 classification, like the results of Table 4 for *top*-1, also show that Softmax obtains superior results for every instance of comparison to *k*-NN, RF, and SVM. Therefore, we summarize them for *top*-1, *top*-2, and *top*-3 considering only network's Softmax classification in Table 5. For a visual summary of both Tables 4 and 5, we refer readers to Fig. 4.

In Table 5, we observe that ConvNeXt demonstrates consistent behavior for the problem, achieving the best or second best result for *top*-1, *top*-2, and *top*-3 classification and for both *image-wise* and *patch-wise classification*. We also observe that SwinTransformer, EfficientNet V2, and VisionTransformer also obtain reasonable results for some instances.

Table 5 also shows that Fusion-FCN presents superior results compared to the baseline networks on a few occasions, which evidences that this fusion approach can improve accuracy.

As a complement, in Figs. 5, 6, and 7, we present the training behavior for ConvNeXt, VisionTransformer, and Fusion-FCN, respectively. In those figures, though we registered the best obtained model during training according to each metric, *i.e.*, Network's loss (LOSS), ACC, and NA, we ended up employing for experimentation and reporting results just the best model obtained according to LOSS, as usual in machine-learning experiments.

As mentioned before, for a summary of both Tables 4 and 5, we refer readers to Fig. 4, where we can observe that *image-wise classification* shows a general superior accuracy compared to the *patch-wise evaluation*. A possible reason for that superiority is the *voting scheme* mentioned in Section 3.2. It also indicates that the entirety of one image might not demonstrate well the main class, *i.e.*, an example can contain parts belonging to a secondary class. We also see in Fig. 4, for both subplots, that the difference in NA among the architectures is small, *i.e.*, network size is not a factor of improvement for the problem. We observe in the second subplot that *top*-3 is much higher than *top*-1 NA, hence, the networks are learning features of the classes, although not able to correctly identify the main one. Interestingly, we also observe that *k*-NN stands as the second best classification method (after the clear outstanding of Softmax) for *image-wise classification*, however, when it comes to *patch-wise classification*, *k*-NN is the worst.

In Fig. 8, we present some examples of correctly and incorrectly classified images with low and high confidence score. The confidence score employed for this selection is based on the number of patches of the image that voted for the classified class. We selected the examples based solely on the first of the eight partitions employed in the experiments.

3.4. Manual classification

Based on the same dataset described in Section 3.1, six members of our research team volunteered to perform the manual classification of an *earlier partition* we defined for the same dataset. The seven *ignored images* of Fig. 1 were included. Those seven examples consist of DCIs instead of DCPs, as the plugs were broken. Furthermore, at the time we accomplished this manual classification, before performing the split specified in Table 2 and described in Section 3.1, we performed an *automatic segmentation* of the dataset, which led us to more "valid" images, counting a total of 1692 instead of the 1499 mentioned in Section 3.1. The dataset was also split into roughly 60%, 16%, and 24% for *training*, *validation*, and *test*, respectively, as described in Section 3.2, consequently, 407 images were considered on the *test* set for this manual classification.

The setup for the manual classification is as follows: Each volunteer received both *training* and *validation* sets with labels and was able to consult those images to check which of the 7 classes each one belongs to. The 407 images comprising the *test* set were given to the volunteers without any label; the filenames of those images were renamed according to an index i; in our case, i.png,[12] in which $i \in [0, 406]$. Then, each volunteer could consult the *training* and *validation* sets while performing the classification of the *test* set. After the 407 images were annotated/classified by each volunteer, the labels were then sent to the corresponding author of this manuscript, and this author automatically calculated the accuracy and informed each volunteer about their performance. In Table 6, we show NA and ACC obtained per volunteer as well as for Inception V3 on the same data partition.

For the results in Table 6, Volunteer 2 performed their classification and later saw the confusion matrix of their performance. Volunteer 2 observed how imbalanced the dataset was and decided to provide new labels for some of the classified images, leading them to the classification shown in that table as Volunteer 2 (2^{nd}). Although classification from Volunteer 2 (2^{nd}) does not fit in a *fair scenario*, we decided to report it here as it shows that knowledge on how imbalanced the *test* set is can guide the classification.

[12] The images of the dataset were originally available in Joint Photographic Experts Group (JPEG) format, though we have re-saved in Portable Network Graphics (PNG) the exact pixel contents loaded from the JPEG originals. Saving the *test* set as new files with *lossless compression* was a step considered to avoid any metadata that could bias volunteers' classification and keep solely the information employed by the networks.

Table 4
Results in terms of NA for employed networks#2. For each network (or *fusion* row) and for each *group of results* (Images or Patches), we mark the greatest result in **bold**.

Network	Images				Patches			
	k-NN	RF	SVM	Softmax	*k*-NN	RF	SVM	Softmax
SqueezeNet	0.621 ± 0.014	0.616 ± 0.011	0.604 ± 0.012	**0.663 ± 0.012**	0.514 ± 0.007	0.539 ± 0.008	0.529 ± 0.009	**0.605 ± 0.007**
MobileNet V2	0.618 ± 0.018	0.596 ± 0.015	0.586 ± 0.015	**0.639 ± 0.009**	0.517 ± 0.008	0.535 ± 0.009	0.528 ± 0.010	**0.587 ± 0.007**
MobileNet V3	0.625 ± 0.011	0.601 ± 0.012	0.584 ± 0.011	**0.651 ± 0.015**	0.510 ± 0.006	0.532 ± 0.007	0.525 ± 0.008	**0.578 ± 0.006**
MNASNet	0.607 ± 0.011	0.602 ± 0.012	0.588 ± 0.012	**0.645 ± 0.007**	0.506 ± 0.006	0.526 ± 0.009	0.519 ± 0.009	**0.584 ± 0.006**
ShuffleNet V2	0.655 ± 0.015	0.640 ± 0.011	0.618 ± 0.010	**0.670 ± 0.012**	0.537 ± 0.007	0.558 ± 0.008	0.550 ± 0.009	**0.604 ± 0.007**
GoogLeNet	0.613 ± 0.012	0.598 ± 0.016	0.583 ± 0.014	**0.653 ± 0.010**	0.491 ± 0.006	0.517 ± 0.007	0.510 ± 0.008	**0.572 ± 0.008**
DenseNet	0.645 ± 0.010	0.629 ± 0.013	0.618 ± 0.009	**0.665 ± 0.009**	0.544 ± 0.007	0.564 ± 0.008	0.558 ± 0.008	**0.609 ± 0.008**
AlexNet	0.624 ± 0.011	0.615 ± 0.011	0.606 ± 0.011	**0.651 ± 0.009**	0.531 ± 0.007	0.552 ± 0.008	0.549 ± 0.009	**0.604 ± 0.008**
ResNet	0.637 ± 0.016	0.602 ± 0.016	0.587 ± 0.013	**0.645 ± 0.008**	0.513 ± 0.007	0.534 ± 0.008	0.527 ± 0.007	**0.585 ± 0.007**
EfficientNet	0.587 ± 0.009	0.550 ± 0.008	0.531 ± 0.011	**0.656 ± 0.013**	0.448 ± 0.004	0.469 ± 0.005	0.462 ± 0.005	**0.537 ± 0.007**
ResNeXt	0.598 ± 0.011	0.591 ± 0.009	0.582 ± 0.010	**0.638 ± 0.013**	0.496 ± 0.007	0.518 ± 0.008	0.510 ± 0.008	**0.569 ± 0.008**
SwinTransformer	0.660 ± 0.009	0.654 ± 0.008	0.649 ± 0.011	**0.679 ± 0.009**	0.566 ± 0.006	0.594 ± 0.007	0.589 ± 0.007	**0.631 ± 0.006**
EfficientNet V2	0.635 ± 0.015	0.625 ± 0.011	0.619 ± 0.013	**0.695 ± 0.010**	0.537 ± 0.007	0.555 ± 0.008	0.549 ± 0.008	**0.610 ± 0.007**
Wide ResNet	0.630 ± 0.010	0.622 ± 0.016	0.599 ± 0.014	**0.650 ± 0.015**	0.513 ± 0.008	0.534 ± 0.008	0.526 ± 0.009	**0.578 ± 0.008**
VGG	0.620 ± 0.005	0.612 ± 0.012	0.602 ± 0.014	**0.666 ± 0.008**	0.527 ± 0.006	0.546 ± 0.008	0.539 ± 0.009	**0.600 ± 0.007**
ConvNeXt	0.657 ± 0.010	0.649 ± 0.012	0.639 ± 0.011	**0.693 ± 0.012**	0.574 ± 0.007	0.600 ± 0.008	0.595 ± 0.008	**0.640 ± 0.008**
VisionTransformer	0.652 ± 0.011	0.642 ± 0.013	0.640 ± 0.013	**0.683 ± 0.009**	0.566 ± 0.008	0.585 ± 0.010	0.578 ± 0.010	**0.629 ± 0.008**
RegNet	0.643 ± 0.011	0.625 ± 0.010	0.613 ± 0.011	**0.680 ± 0.009**	0.536 ± 0.006	0.554 ± 0.007	0.546 ± 0.008	**0.599 ± 0.006**
Mean	0.629 ± 0.012	0.615 ± 0.012	0.603 ± 0.012	**0.662 ± 0.010**	0.524 ± 0.007	0.545 ± 0.008	0.538 ± 0.008	**0.596 ± 0.007**
Fusion	0.667 ± 0.010	0.672 ± 0.011	0.649 ± 0.013	**0.688 ± 0.010**	0.589 ± 0.009	0.613 ± 0.010	0.603 ± 0.010	**0.651 ± 0.009**

In Table 6, we observe that Inception V3 obtains better NA than most of the volunteers. In fact, just Volunteer 2 (2nd) obtained higher NA compared to this network. Even if we consider ACC, Inception V3 obtains worse accuracy only when compared to Volunteer 2 (2nd) and Volunteer 4, showing superiority in terms of this traditional metric when compared to Volunteer 1, Volunteer 2, Volunteer 3, Volunteer 5, and Volunteer 6.

Results in Table 6, mainly when compared to the results presented in Tables 4 and 5 through Section 3.3, evidence the difficulty of the problem for visually distinguishing among the classes, as the manual classification accuracy tend to be lower than those obtained through machine-learning methods. Recalling that Fig. 3 also reveals this difficulty through the high variability of patch appearances within each class.

In this classification, there were four examples for which all volunteers correctly classified them while the network did not. Also, there were four other examples for which the network correctly classify them while all volunteers got them wrong (although not agreeing in the wrong label). For illustrative purpose, we present in Fig. 9 such examples.

4. Key conclusions and recommendations

In this manuscript, we have presented an extensive overview of the literature on rock-type classification focusing on the use of machine-learning techniques/architectures. We observed a wide range of classification results obtained in the literature (shown in Table 1), which can likely be attributed to the fact that these studies not only use different datasets for experimentation but also other types of images, *e.g.*, DCP, BWI, DCI, WL, BEI, TSI, xCT, and NRI. Throughout this literature review, we pointed out common practices of a number of studies (*e.g.*, many tend to employ well-known network architectures) and common pitfalls on the machine-learning experimental setup (*e.g.*, data contamination).

Due to the lack of a standard benchmark for the problem, the alternatives in the literature are difficult to compare. Also, influence on the inconclusiveness for the best approach show the lack of an extensive comparison with the state-of-the-art machine-learning techniques by those works in the literature. In fact, most of them tend to present results just for the employed method, being it based on traditional machine-learning methods (*e.g.*, SVMs, Bayes' rule), a hand-crafted CNN, or a CNN whose architecture have been proposed and consolidated in the machine-learning literature.

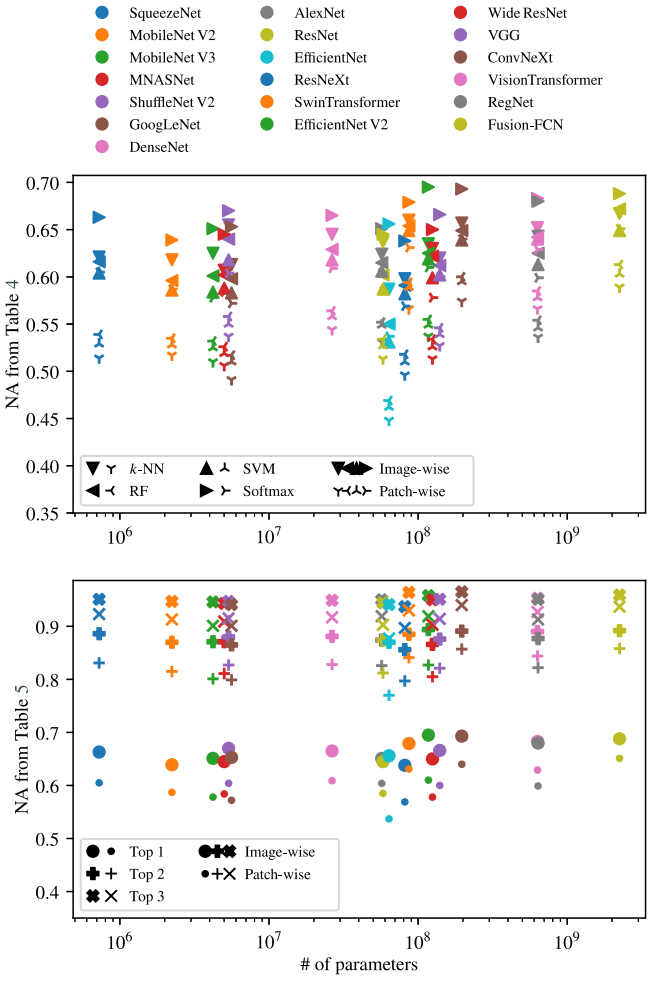


Fig. 4. Visual summary of results. Subplots comprise data from Tables 4 and 5, respectively. Networks are organized by color and sorted (from top to bottom and from left to right in the legend) by their number of parameters, as in Table 3.

For the DCP classification problem for which we have performed experiments (reported in Section 3.3), we showed that although the

Table 5
Results in terms of NA considering *top* 1–3 classification. Results are for straight network classification, a.k.a., Softmax classification. For each column, we mark the greatest and second greatest results in **bold** and ***italic bold***, respectively. In *Fusion* row, results are marked in **bold** when fusion gets better results than the best result on the respective column.

Network	Images			Patches		
	Top 1	Top 2	Top 3	Top 1	Top 2	Top 3
SqueezeNet	0.663 ± 0.012	0.886 ± 0.007	0.951 ± 0.004	0.605 ± 0.007	0.831 ± 0.004	0.923 ± 0.003
MobileNet V2	0.639 ± 0.009	0.870 ± 0.008	0.947 ± 0.004	0.587 ± 0.007	0.815 ± 0.006	0.913 ± 0.003
MobileNet V3	0.651 ± 0.015	0.871 ± 0.005	0.946 ± 0.004	0.578 ± 0.006	0.801 ± 0.004	0.901 ± 0.003
MNASNet	0.645 ± 0.007	0.871 ± 0.009	0.943 ± 0.003	0.584 ± 0.006	0.811 ± 0.005	0.909 ± 0.003
ShuffleNet V2	0.670 ± 0.012	0.880 ± 0.003	0.947 ± 0.007	0.604 ± 0.007	0.827 ± 0.005	0.915 ± 0.005
GoogLeNet	0.653 ± 0.010	0.865 ± 0.005	0.941 ± 0.005	0.572 ± 0.008	0.799 ± 0.005	0.901 ± 0.004
DenseNet	0.665 ± 0.009	0.881 ± 0.007	0.949 ± 0.005	0.609 ± 0.008	0.828 ± 0.006	0.917 ± 0.004
AlexNet	0.651 ± 0.009	0.874 ± 0.006	0.950 ± 0.007	0.604 ± 0.008	0.826 ± 0.005	0.919 ± 0.003
ResNet	0.645 ± 0.008	0.874 ± 0.007	0.944 ± 0.007	0.585 ± 0.007	0.812 ± 0.005	0.903 ± 0.005
EfficientNet	0.656 ± 0.013	0.870 ± 0.010	0.941 ± 0.007	0.537 ± 0.007	0.770 ± 0.005	0.877 ± 0.004
ResNeXt	0.638 ± 0.013	0.856 ± 0.007	0.937 ± 0.006	0.569 ± 0.008	0.797 ± 0.006	0.897 ± 0.006
SwinTransformer	0.679 ± 0.009	0.885 ± 0.007	***0.964 ± 0.004***	***0.631 ± 0.006***	0.841 ± 0.004	***0.930 ± 0.003***
EfficientNet V2	**0.695 ± 0.010**	**0.894 ± 0.004**	0.958 ± 0.004	0.610 ± 0.007	0.827 ± 0.005	0.919 ± 0.005
Wide ResNet	0.650 ± 0.015	0.866 ± 0.007	0.950 ± 0.004	0.578 ± 0.008	0.805 ± 0.006	0.903 ± 0.005
VGG	0.666 ± 0.008	0.876 ± 0.005	0.951 ± 0.006	0.600 ± 0.007	0.821 ± 0.005	0.914 ± 0.004
ConvNeXt	***0.693 ± 0.012***	***0.891 ± 0.006***	**0.965 ± 0.003**	**0.640 ± 0.008**	**0.857 ± 0.004**	**0.940 ± 0.003**
VisionTransformer	0.683 ± 0.009	0.890 ± 0.005	0.953 ± 0.006	0.629 ± 0.008	***0.844 ± 0.004***	0.927 ± 0.003
RegNet	0.680 ± 0.009	0.877 ± 0.006	0.951 ± 0.008	0.599 ± 0.006	0.822 ± 0.004	0.913 ± 0.004
Mean	0.662 ± 0.010	0.876 ± 0.006	0.949 ± 0.005	0.596 ± 0.007	0.818 ± 0.005	0.912 ± 0.004
Fusion	0.688 ± 0.010	0.892 ± 0.006	0.959 ± 0.004	**0.651 ± 0.009**	**0.858 ± 0.004**	0.937 ± 0.003

Table 6
Manual classification performed by volunteers. For each column, we mark the greatest and second greatest results in **bold** and ***italic bold***, respectively. The row for Volunteer 2 (2^{nd}) refers to the reworked classification (the second) of Volunteer 2 after observing how imbalanced the dataset was through the confusion matrix of their own first classification.

	NA	ACC
Volunteer 1	0.573	0.543
Volunteer 2	0.555	0.538
Volunteer 2 (2^{nd})	**0.680**	**0.671**
Volunteer 3	0.555	0.538
Volunteer 4	0.670	***0.649***
Volunteer 5	0.599	0.479
Volunteer 6	0.492	0.464
Inception V3	***0.671***	0.599

accuracy for *top*-1 classification is low (average of 66.2% in terms of NA), the correct class is returned by the classification methods among the *top*-2 and *top*-3 predictions in 87.6% and 94.9%, respectively, in terms of NA, which proves that the correct class can be learned by the networks despite the inherent confusion among them. Furthermore, our manual classification of DCPs (presented in Section 3.4) also confirms this *inter-class confusion*.

4.1. Difficulties of the classification problem

We presented an extensive comparison of state-of-the-art classification networks comprising most of the well-known CNN architectures and the recently proposed Vision Transformer (ViT) (Dosovitskiy et al., 2021) approach. For the 7-class DCP classification problem we experimented with, we obtained similar accuracy among the network alternatives (with just slightly improved results by the ConvNeXt architecture), which indicates we are close to the maximum classification accuracy for the problem, a.k.a., the *Bayes's error* (Bishop, 2006).

We previously mentioned that three of the difficulties in classifying DCPs are the *label non-agreement*, the *intra-class dissimilarity*, and the *inter-class similarity*. One fact that raises those problems is that a single DCP can contain samples from many rock types — especially for carbonate reservoirs —, as has been shown by Mohamed et al. (2011) in their work analyzing *reservoir heterogeneity*.

4.2. Recommendations and future work

As the problem of DCP classification suffers from *label non-agreement* — as well as classification problems based on other types of data —, future work can be accomplished by considering multiple *ground-truth* for the dataset, *i.e.*, different annotations by expert geologists. Furthermore, enhancing the accuracy of rock-type classification necessitates advancing interpretability studies and integrating geological knowledge into the development process.

We also observed that many authors have been employing well-known network architectures for diverse rock-classification problems. Most of those networks, however, are *general methods* for *computer vision*. As many of those rock-classification problems present fine-grained similarities and dissimilarities among the classes, they can benefit from networks specially tailored for fine-grained recognition, as is the case of the *bilinear CNNs* introduced by Lin et al. (2017).

We advise authors dealing with rock classification problems to pay close attention to the considered *experimental setup* so as to avoid any kind of data contamination. It is common that geologists usually acquire multiple images of a single rock sample so, when devising the machine-learning experimental setup, it is important to ensure that all data (images, for instance) referring to the same rock sample fall into the same set — *training*, *validation*, or *test*. Also, when a *patch-based approach* is employed, it is essential that all patches extracted from the same image also fall into a single *training*, *validation*, or *test* set. Furthermore, in case multiple images are acquired for the same rock sample, it is necessary to ensure the patches referring to the same rock sample fall into the same partitioned set.

Labeling images and obtaining the actual rock types of examples is a time-consuming task, which makes it prone to errors. As mentioned, rock-type classification suffers from the problem of *label non-agreement*, which indicate that *clustering analysis* can benefit the area by providing information about which examples are similar to one another, thus aiding specialists in reviewing their own labels for a dataset being prepared.

Finally, a future work that can be accomplished is for preparing and publishing a benchmark which the community can use for experimentation. It would facilitate comparison among the methods and help developing the state of the art for the problem. To serve as a benchmark for machine-learning experiments for DCP, it will be necessary to avoid the data contamination problem we discussed along this manuscript when preparing *training*, *validation* and *test* sets, *i.e.*, to ensure that

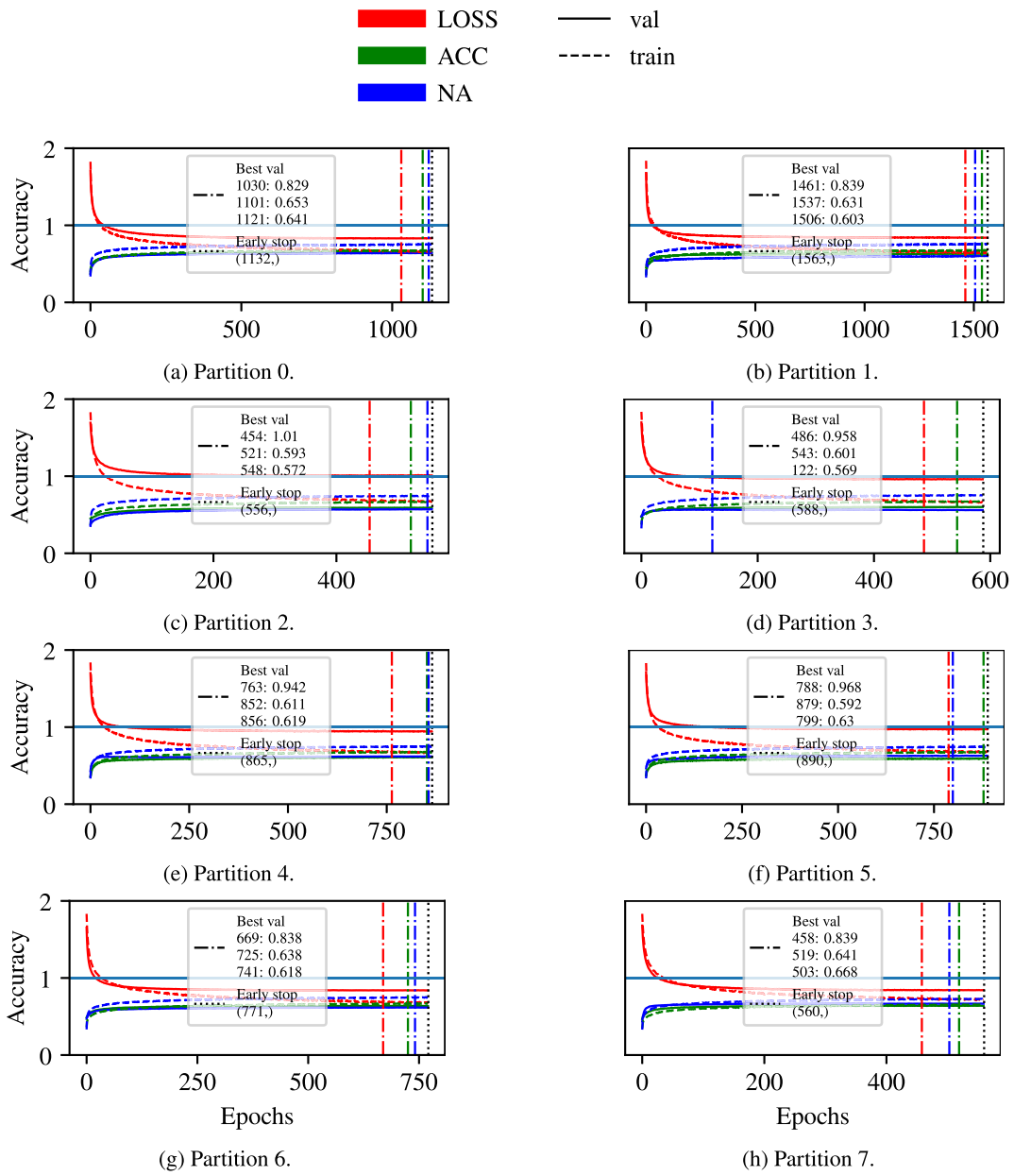


Fig. 5. Training behavior for ConvNeXt. The legend for curve colors and types is at the top. The colors of the legend also apply to the "Best val" vertical lines specifying the best model on validation according to each metric (color). The three "A: B" lines in "Best val" associates with LOSS, ACC, and NA, respectively, in which "A" indicates the epoch with the best respective metric and "B" indicates the respective metric obtained on the *validation* set.

all examples coming from the same rock sample fall into the same set. Some datasets already available online, *e.g.*, ODP, GSSA, and CGSI provide no preparation of the data for machine-learning experiments and it happens that authors might split it in such a way that image crops from the same rock sample fall into *training* and *test* sets, which lead to a data contamination problem. In practice, when classifying a new DC, none of its pixels will have been used for training the classifier (which is the scenario that must be reproduced in the data preparation and experimental setup).

CRediT authorship contribution statement

Pedro Ribeiro Mendes Júnior: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Soroor Salavati:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Oscar Linares:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Maiara Moreira Gonçalves:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Marcelo Ferreira Zampieri:** Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft. **Vitor Hugo de Sousa Ferreira:** Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft. **Manuel Castro:** Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft. **Rafael de Oliveira Werneck:** Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Renato Moura:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Elayne Morais:** Data curation, Investigation, Methodology, Software, Validation, Writing – original draft. **Ahmed Esmin:** Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization. **Leopoldo**

Lusquino Filho: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Writing – original draft. **Denis José Schiozer:** Funding acquisition, Project administration, Supervision. **Alexandre Ferreira:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **Alessandra Davólio:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Software, Supervision, Validation, Writing – original draft. **Anderson Rocha:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

Code availability section

Rock classification

Contact: pedrormjunior@gmail.com

Hardware requirements: 32 GB of main memory and GPUs (optional).

Program language: Python

Software required: See `environment.yaml` file.

Program size: 352KB

The source codes are available for downloading at the link: https://github.com/pedrormjunior/rock-classification-cageo.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pedro Ribeiro Mendes Junior reports financial support was provided by Shell Brazil Oil. Soroor Salavati reports financial support was provided by Shell Brazil Oil. Oscar Cuadros Linares reports financial support

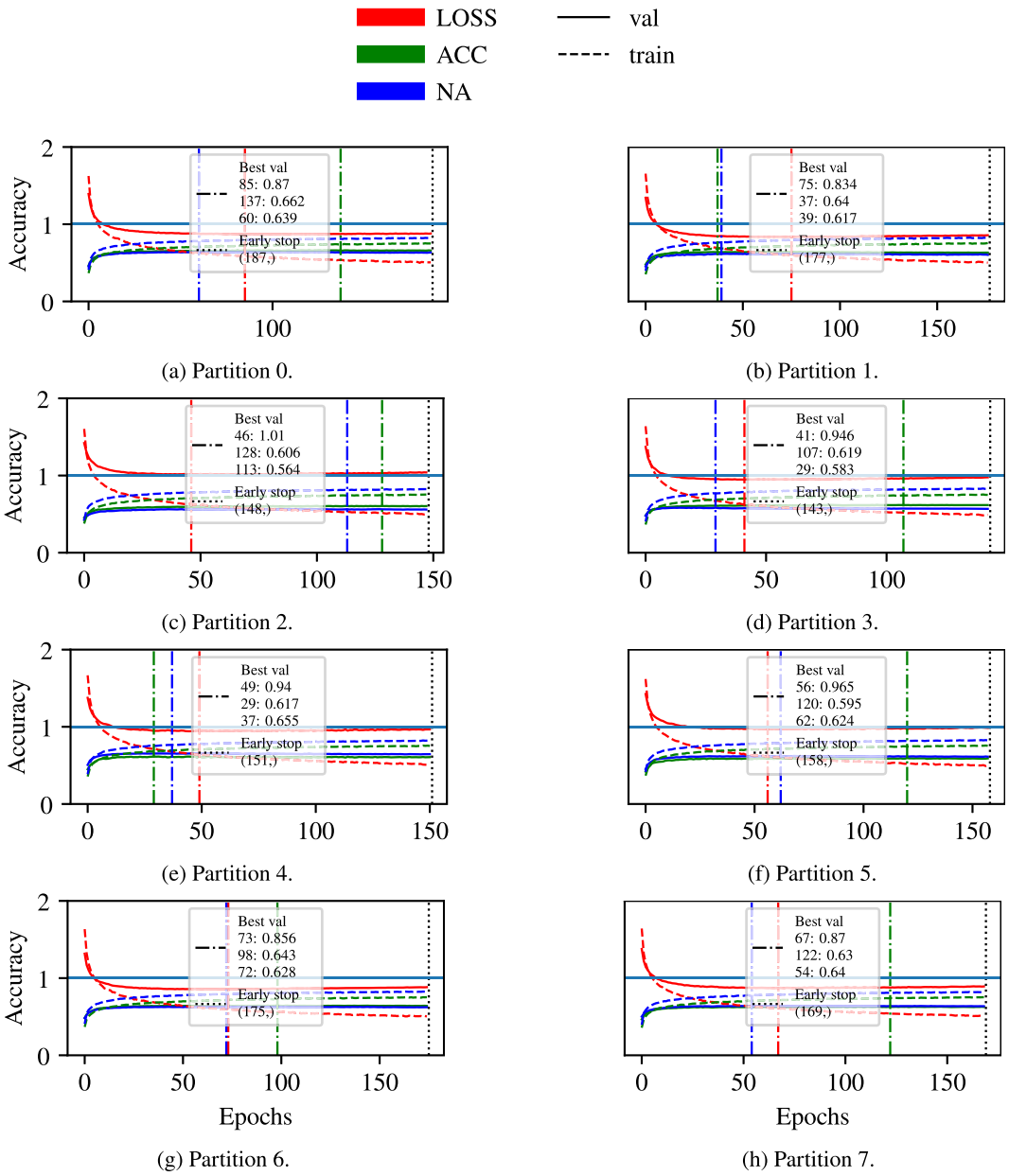


Fig. 6. Training behavior for VisionTransformer. The legend for curve colors and types is at the top. The colors of the legend also apply to the "Best val" vertical lines specifying the best model on validation according to each metric (color). The three "A: B" lines in "Best val" associates with LOSS, ACC, and NA, respectively, in which "A" indicates the epoch with the best respective metric and "B" indicates the respective metric obtained on the *validation* set.

was provided by Shell Brazil Oil. Maiara Moreira Goncalves reports financial support was provided by Shell Brazil Oil. Marcelo Ferreira Zampieri reports financial support was provided by Shell Brazil Oil. Vitor Hugode Sousa Ferreira reports financial support was provided by Shell Brazil Oil. Manuel Castro reports financial support was provided by Shell Brazil Oil. Rafael de Oliveira Werneck reports financial support was provided by Shell Brazil Oil. Renato Moura reports financial support was provided by Shell Brazil Oil. Elayne Morais reports financial support was provided by Shell Brazil Oil. Ahmed Esmin reports financial support was provided by Shell Brazil Oil. Leopoldo Lusquino Filho reports financial support was provided by Shell Brazil Oil. Denis Jose Schiozer reports financial support was provided by Shell Brazil Oil. Alexandre Ferreira reports financial support was provided by Shell Brazil Oil. Alessandra Davolio reports financial support was provided by Shell Brazil Oil. Anderson Rocha reports financial support was provided by Shell Brazil Oil.

Acknowledgments

This research was carried out as part of the finished R&D project with Shell Brasil Petróleo Ltda, registered as ANP number 21373-6 as "Machine-Learning Development for Analysis of Complex Production Data in a Pre-Salt Carbonate Field" (UNICAMP/Shell Brazil/ANP) and of the ongoing R&D project with Shell Brasil Petróleo Ltda, registered as ANP number 24282-6 as "Artificial intelligence and machine learning for analyzing complex and multispectral production data, establishing causal links, effects of interventions and hybrid modeling in Pre-Salt fields – Phase 2" (UNICAMP/Shell Brazil/ANP), both of them funded by Shell Brazil Technology, under the ANP R&D levy as "Compromisso de Investimentos com Pesquisa e Desenvolvimento". We thank Shell Brazil for permission to publish this work.

Finally, we thank all developers for the essential tools we employed for accomplishing this work (Hunter, 2007; Perez and Granger, 2007; Pedregosa et al., 2011, etc.).

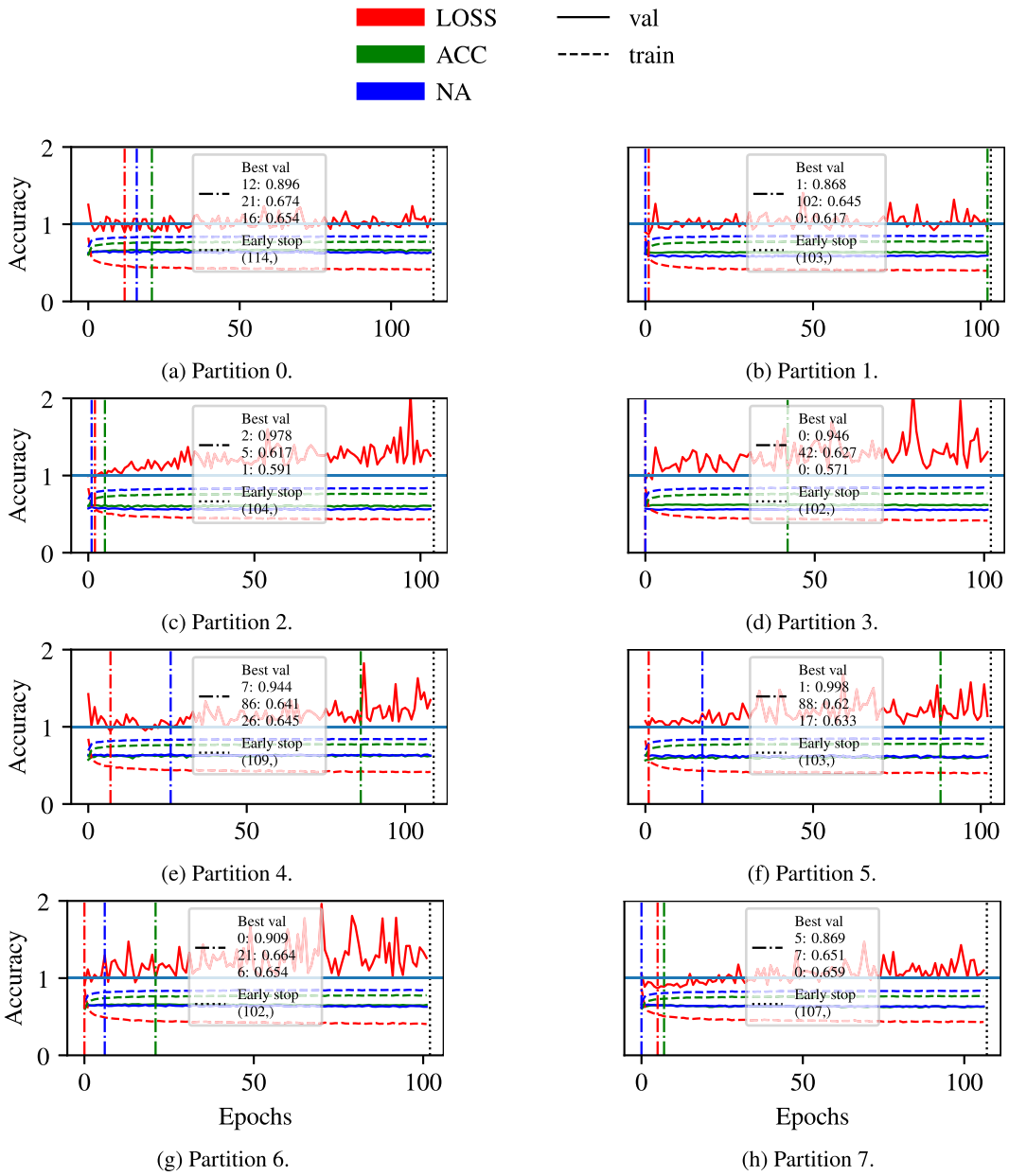
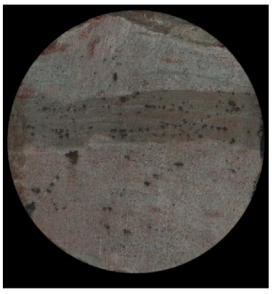


Fig. 7. Training behavior for Fusion-FCN. The legend for curve colors and types is at the top. The colors of the legend also apply to the "Best val" vertical lines specifying the best model on validation according to each metric (color). The three "A: B" lines in "Best val" associates with LOSS, ACC, and NA, respectively, in which "A" indicates the epoch with the best respective metric and "B" indicates the respective metric obtained on the *validation* set.

(a) Grainstones/Rudstones.

(b) Calcimudstones.



(c) Shales.

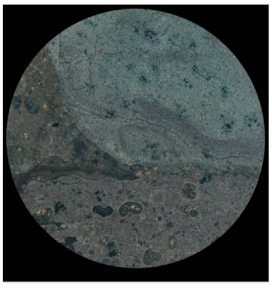
(d) Crystal shrubs.

(e) Grainstones/Rudstones.

(f) Grainstones/Rudstones.

(g) Correct: Crystal shrubs. Incorrect: Grainstones/Rudstones.

(h) Correct: Grainstones/Rudstones. Incorrect: Conglomerates.



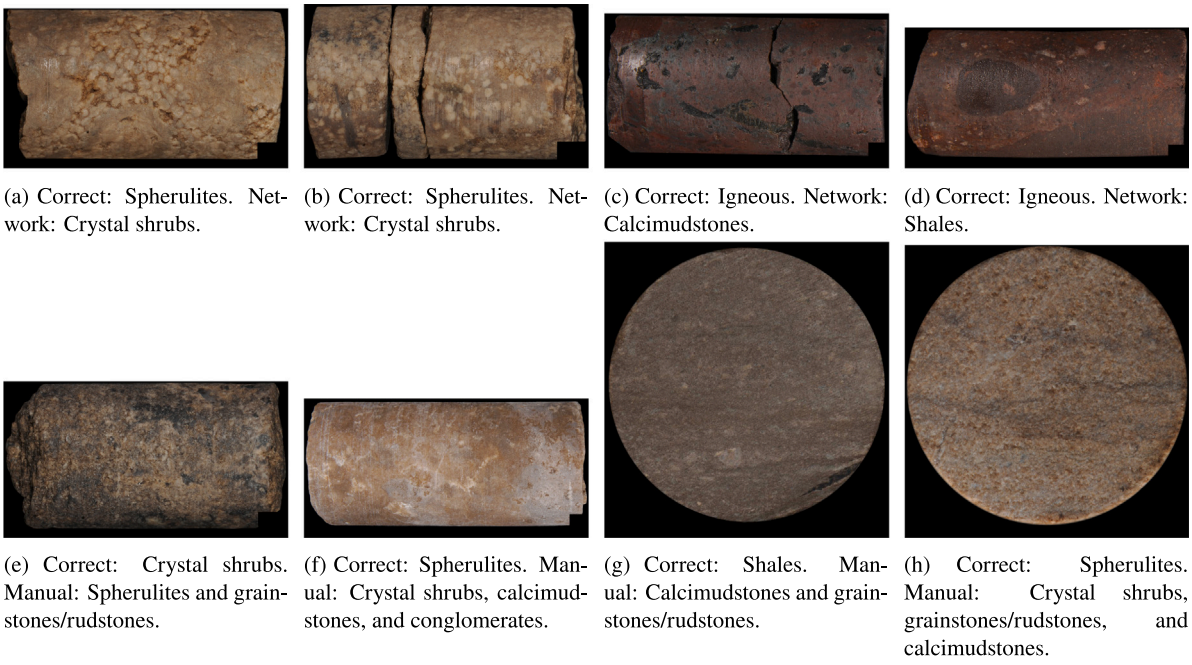
(i) Correct: Conglomerates. Incorrect: Shales.

(j) Correct: Spherulites. Incorrect: Conglomerates.

(k) Correct: Grainstones/Rudstones. Incorrect: Conglomerates.

(l) Correct: Spherulites. Incorrect: Calcimudstones.

Fig. 8. Examples of correctly and incorrectly classified images with low and high confidence score. (a–c) Correctly classified with low score. (d–f) Correctly classified with high score. (g–i) Incorrectly classified with low score. (j–l) Incorrectly classified with high score.



(a) Correct: Spherulites. Network: Crystal shrubs.

(b) Correct: Spherulites. Network: Crystal shrubs.

(c) Correct: Igneous. Network: Calcimudstones.

(d) Correct: Igneous. Network: Shales.

(e) Correct: Crystal shrubs. Manual: Spherulites and grainstones/rudstones.

(f) Correct: Spherulites. Manual: Crystal shrubs, calcimudstones, and conglomerates.

(g) Correct: Shales. Manual: Calcimudstones and grainstones/rudstones.

(h) Correct: Spherulites. Manual: Crystal shrubs, grainstones/rudstones, and calcimudstones.

Fig. 9. Examples correctly classified by volunteers only and by network only. (a–d) Correctly classified by all volunteers. (e–h) Correctly classified only by the network.

Data availability

The authors do not have permission to share data.

References

Al-Mudhafar, W.J., 2017. Integrating well log interpretations for lithofacies classification and permeability modeling through advanced machine learning algorithms. J. Petrol. Explor. Prod. Technol. 7 (4), 1023–1033. http://dx.doi.org/10.1007/s13202-017-0360-0.

Almisned, O.A., Alqahtani, N.B., 2021. Rock analysis to characterize Saudi soft sandstone rock. J. Petrol. Explor. Prod. Technol. 11 (6), 2381–2387. http://dx.doi.org/10.1007/s13202-021-01160-y.

Alzubaidi, F., Mostaghimi, P., Swietojanski, P., Clark, S.R., Armstrong, R.T., 2021. Automated lithology classification from drill core images using convolutional neural networks. Elsevier J. Petrol. Sci. Eng. 197, 107933. http://dx.doi.org/10.1016/j.petrol.2020.107933.

Andrä, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., Marsh, M., Mukerji, T., Saenger, E.H., Sain, R., Saxena, N., Ricker, S., Wiegmann, A., Zhan, X., 2013. Digital rock physics benchmarks—Part I: Imaging and segmentation. Elsevier Comput. Geosci. 50, 25–32. http://dx.doi.org/10.1016/j.cageo.2012.09.005.

Badr, A.R., Ayoub, M.R., 1989. Study of a complex carbonate reservoir using the formation MicroScanner (FMS) tool. In: Middle East Oil Show. Bahrain Exhibition Centre, pp. 507–516. http://dx.doi.org/10.2118/17977-ms.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495. http://dx.doi.org/10.1109/tpami.2016.2644615.

Baraboshkin, E.E., Ismailova, L.S., Orlov, D.M., Zhukovskaya, E.A., Kalmykov, G.A., Khotylev, O.V., Baraboshkin, E.Y., Koroteev, D.A., 2020. Deep convolutions for in-depth automated rock typing. Elsevier Comput. Geosci. 135, 104330. http://dx.doi.org/10.1016/j.cageo.2019.104330.

Bestagini, P., Lipari, V., Tubaro, S., 2017. A machine learning approach to facies classification using well logs. In: SEG Technical Program Expanded Abstracts 2017. pp. 2137–2142. http://dx.doi.org/10.1190/segam2017-17729805.1.

Beucher, S., 1979. Use of watersheds in contour detection. In: International Workshop on Image Processing. Rennes, France, pp. 17–21, URL: https://www.researchgate.net/publication/230837989.

Bishop, C.M., 2006. Pattern recognition and machine learning, first ed. In: Information Science and Statistics, Springer-Verlag New York, URL: https://www.springer.com/us/book/9780387310732.

Boulahia, S.Y., Amamra, A., Madi, M.R., Daikh, S., 2021. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. Mach. Vis. Appl. 32 (6), http://dx.doi.org/10.1007/s00138-021-01249-8.

Breiman, L., 2001. Random forests. Springer Mach. Learn. 45 (1), 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Budennyy, S., Pachezhertsev, A., Bukharev, A., Erofeev, A., Mitrushkin, D., Belozerov, B., 2017. Image processing and machine learning approaches for petrographic thin section analysis. In: SPE Russian Petroleum Technology Conference. Moscow, Russia, pp. 1–12. http://dx.doi.org/10.2118/187885-ms.

Caja, M.A., Peña, A.C., Campos, J.R., Diego, L.G., Tritlla, J., Bover-Arnal, T., Martín-Martín, J.D., 2019. Image processing and machine learning applied to lithology identification, classification and quantification of thin section cutting samples. In: SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers, Calgary, Alberta, Canada, pp. 1–8. http://dx.doi.org/10.2118/196117-ms.

Chatterjee, S., 2012. Vision-based rock-type classification of limestone using multi-class support vector machine. Springer Appl. Intell. 39 (1), 14–27. http://dx.doi.org/10.1007/s10489-012-0391-7.

Chen, M.Y., Dahan, C.A., Ekstrom, M.P., Lloyd, P.M., Rossi, D.J., 1987. Formation imaging with microelectrical scanning arrays. Log Anal. 28 (3), URL: https://onepetro.org/petrophysics/article-abstract/171858.

Cheng, G., Guo, W., 2017. Rock images classification by using deep convolution neural network. J. Phys. Conf. Ser. 887, 012089. http://dx.doi.org/10.1088/1742-6596/887/1/012089.

Coates, A., Ng, A.Y., 2012. Learning feature representations with K-means. In: Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade, second ed. In: Lecture Notes in Computer Science, vol. 7700, Springer, Berlin, Heidelberg, pp. 561–580. http://dx.doi.org/10.1007/978-3-642-35289-8_30.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Springer Mach. Learn. 20 (3), 273–297. http://dx.doi.org/10.1007/BF00994018.

Dakhelpour-Ghoveifel, J., Shegeftfard, M., Dejam, M., 2018. Capillary-based method for rock typing in transition zone of carbonate reservoirs. J. Petrol. Explor. Prod. Technol. 9 (3), 2009–2018. http://dx.doi.org/10.1007/s13202-018-0593-6.

Deng, Z., Cao, M., Rai, L., Gao, W., 2018. A two-stage classification method for borehole-wall images with support vector machine. PLoS One 13 (6), e0199749. http://dx.doi.org/10.1371/journal.pone.0199749.

Division of Marine Large Programs, 2023. ODP Leg 197 - Hole 1203A. Online. URL: https://mlp.ldeo.columbia.edu/data/odp/leg197/1203A.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16 × 16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations. pp. 1–21, Virtual. URL: https://iclr.cc/virtual/2021/oral/3458.

Ekstrom, M.P., Dahan, C.A., Chen, M.Y., Lloyd, P.M., Rossi, D.J., 1986. Formation imaging with microelectrical scanning arrays. In: SPWLA Annual Logging Symposium. Houston, TX, USA, URL: https://onepetro.org/SPWLAALS/proceedings-abstract/SPWLA-1986/All-SPWLA-1986/18600, SPWLA-1986-BB.

Folk, R.L., 1959. Practical petrographic classification of limestones. AAPG Bull. 43 (1), 1–38. http://dx.doi.org/10.1306/0bda5c36-16bd-11d7-8645000102c1865d.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (5), 1189–1232, URL: https://www.jstor.org/stable/2699986.

Fu, D., Su, C., Wang, W., Yuan, R., 2022. Deep learning based lithology classification of drill core images. PLoS One 17 (7), e0270826. http://dx.doi.org/10.1371/journal.pone.0270826.

Gill, R., 2023. Geosec slides. Online. URL: https://geosecslides.blogspot.com.

Gomes, J., Bunevich, R., Tedeschi, L., Tucker, M., Whitaker, F., 2020. Facies classification and patterns of lacustrine carbonate deposition of the Barra Velha Formation, Santos Basin, Brazilian Pre-salt. Elsevier Mar. Petrol. Geol. 113, 104176. http://dx.doi.org/10.1016/j.marpetgeo.2019.104176.

Günther, C., Jansson, N., Liwicki, M., Simistira-Liwicki, F., 2021. Towards a machine learning framework for drill core analysis. In: Swedish Artificial Intelligence Society Workshop. Luleå, Sweden, p. 6. http://dx.doi.org/10.1109/sais53221.2021.9484025.

Guojian, C., Peisong, L., 2021. Rock thin-section image classification based on residual neural network. In: IEEE International Conference on Intelligent Computing and Signal Processing. Xi'an, China, pp. 521–524. http://dx.doi.org/10.1109/icsp51882.2021.9408983.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE 67 (5), 786–804. http://dx.doi.org/10.1109/proc.1979.11328.

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Trans. Syst. Man Cybern. SMC-3 (6), 610–621. http://dx.doi.org/10.1109/tsmc.1973.4309314.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE International Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, pp. 770–778. http://dx.doi.org/10.1109/cvpr.2016.90.

Hébert, V., Porcher, T., Planes, V., Léger, M., Alperovich, A., Goldluecke, B., Rodriguez, O., Youssef, S., 2020. Digital core repository coupled with machine learning as a tool to classify and assess petrophysical rock properties. E3S Web Conf. 146, 01003:1–11. http://dx.doi.org/10.1051/e3sconf/202014601003.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., Le, Q., 2019. Searching for MobileNetV3. In: IEEE International Conference on Computer Vision. Seoul, Korea, pp. 1314–1324. http://dx.doi.org/10.1109/iccv.2019.00140.

Huang, G., Liu, Z., Der Maaten, L.V., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: IEEE International Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA, pp. 4700–4708. http://dx.doi.org/10.1109/cvpr.2017.243.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 9 (3), 90–95. http://dx.doi.org/10.1109/MCSE.2007.55, URL: https://matplotlib.org.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size CoRR abs/1602.07360. pp. 1–13, arXiv:1602.07360.

Karimpouli, S., Tahmasebi, P., 2019. Segmentation of digital rock images using deep convolutional autoencoder networks. Elsevier Comput. Geosci. 126, 142–150. http://dx.doi.org/10.1016/j.cageo.2019.02.003.

Katoch, S., Chauhan, S.S., Kumar, V., 2020. A review on genetic algorithm: past, present, and future. Springer Multimedia Tools Appl. 80 (5), 8091–8126. http://dx.doi.org/10.1007/s11042-020-10139-6.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), In: Advances in Neural Information Processing Systems, vol. 25, Curran Associates, Inc., Lake Tahoe, NV, USA, pp. 1097–1105, URL: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Leal F., J.A., Ochoa G., L.H., Contreras F., C.C., 2018. Automatic identification of calcareous lithologies using support vector machines, borehole logs and fractal dimension of borehole electrical imaging. Earth Sci. Res. J. 22 (2), 75–82. http://dx.doi.org/10.15446/esrj.v22n2.68320.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324. http://dx.doi.org/10.1109/5.726791.

Lepistö, L., Kunttu, I., Visa, A., 2005. Rock image classification using color features in Gabor space. J. Electron. Imaging 14 (4), 040503. http://dx.doi.org/10.1117/1.2149872.

Lin, T.Y., RoyChowdhury, A., Maji, S., 2017. Bilinear CNNs for fine-grained visual recognition CoRR abs/1504.07889. pp. 1–14, arXiv:1504.07889.

Linek, M., Jungmann, M., Berlage, T., Pechnig, R., Clauser, C., 2007. Rock classification based on resistivity patterns in electrical borehole wall images. J. Geophys. Eng. 4 (2), 171–183. http://dx.doi.org/10.1088/1742-2132/4/2/006.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE International Conference on Computer Vision. pp. 9992–10002. http://dx.doi.org/10.1109/iccv48922.2021.00986, Virtual.

Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A ConvNet for the 2020s. In: IEEE International Conference on Computer Vision and Pattern Recognition. New Orleans, LA, USA, pp. 11966–11976. http://dx.doi.org/10.1109/cvpr52688.2022.01167.

Ma, N., Zhang, X., Zheng, H.T., Sun, J., 2018. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), European Conference on Computer Vision. In: Lecture Notes in Computer Science, vol. 11205, Springer, Munich, Germany, pp. 122–138. http://dx.doi.org/10.1007/978-3-030-01264-9_8.

Magar, I., Schwartz, R., 2022. Data contamination: From memorization to exploitation, CoRR abs/2203.08242. pp. 1–9, arXiv:2203.08242.

Mao, K., Tan, K.C., Ser, W., 2000. Probabilistic neural-network structure determination for pattern classification. IEEE Trans. Neural Netw. 11 (4), 1009–1016. http://dx.doi.org/10.1109/72.857781.

Mohamed, S.S.E.D., Dernaika, M., Kalam, M.Z., 2011. The impact of heterogeneity and multi-scale measurements on reservoir characterization and STOOIP estimations. In: SPE Reservoir Characterisation and Simulation Conference and Exhibition. Abu Dhabi, UAE, pp. 1–12. http://dx.doi.org/10.2118/147950-ms.

Pascual, A.D.P., Shu, L., Szoke-Sieswerda, J., McIsaac, K., Osinski, G., 2019. Towards natural scene rock image classification with convolutional neural networks. In: IEEE Canadian Conference of Electrical and Computer Engineering. Edmonton, AB, Canada, p. 4. http://dx.doi.org/10.1109/ccece.2019.8861885.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 32, no. 12, Curran Associates, Inc., Vancouver, Canada, URL: https://papers.nips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Paullada, A., Raji, I.D., Bender, E.M., Denton, E., Hanna, A., 2021. Data and its (dis)contents: A survey of dataset development and use in machine learning research. Patterns 2 (11), 100336. http://dx.doi.org/10.1016/j.patter.2021.100336.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830, URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

Perez, F., Granger, B.E., 2007. IPython: A system for interactive scientific computing. Comput. Sci. Eng. 9 (3), 21–29. http://dx.doi.org/10.1109/MCSE.2007.53, URL: https://ipython.org.

Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollar, P., 2020. Designing network design spaces. In: IEEE International Conference on Computer Vision and Pattern Recognition. pp. 10425–10433. http://dx.doi.org/10.1109/cvpr42600.2020.01044, Virtual.

Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y., 2007. Self-taught learning: Transfer learning from unlabeled data. In: International Conference on Machine Learning. ACM Press, Corvallis, OR, USA, pp. 759–766. http://dx.doi.org/10.1145/1273496.1273592.

Ran, X., Xue, L., Zhang, Y., Liu, Z., Sang, X., He, J., 2019. Rock classification from field image patches analyzed using a deep convolutional neural network. Mathematics 7 (8), 755. http://dx.doi.org/10.3390/math7080755.

Rocha, A., Goldenstein, S., 2014. Multiclass from binary: Expanding one-vs-all, one-vs-one and ECOC-based approaches. IEEE Trans. Neural Netw. Learn. Syst. 25 (2), 289–302. http://dx.doi.org/10.1109/TNNLS.2013.2274735.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. Springer Int. J. Comput. Vis. 115 (3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In: IEEE International Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, pp. 4510–4520. http://dx.doi.org/10.1109/cvpr.2018.00474.

Sharif, H., Ralchenko, M., Samson, C., Ellery, A., 2015. Autonomous rock classification using Bayesian image analysis for rover-based planetary exploration. Elsevier Comput. Geosci. 83, 153–167. http://dx.doi.org/10.1016/j.cageo.2015.05.011.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6 (1), http://dx.doi.org/10.1186/s40537-019-0197-0.

Shu, L., McIsaac, K., Osinski, G.R., Francis, R., 2017. Unsupervised feature learning for autonomous rock image classification. Elsevier Comput. Geosci. 106, 10–17. http://dx.doi.org/10.1016/j.cageo.2017.05.010.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. San Juan, Puerto Rico, pp. 1–14, URL: https://arxiv.org/abs/1409.1556.

Su, C., Xu, S.j., Zhu, K.y., Zhang, X.c., 2020. Rock classification in petrographic thin section images based on concatenated convolutional neural networks. Earth Sci. Inform. 13 (4), 1477–1484. http://dx.doi.org/10.1007/s12145-020-00505-1.

Sudakov, O., Burnaev, E., Koroteev, D., 2019. Driving digital rock towards machine learning: Predicting permeability with gradient boosting and deep neural networks. Elsevier Comput. Geosci. 127, 91–98. http://dx.doi.org/10.1016/j.cageo.2019.02.002.

Sutskever, I., Martens, J., Dahl, G., Hinton, G., 2013. On the importance of initialization and momentum in deep learning. In: Dasgupta, S., McAllester, D. (Eds.), International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 28, no. 3, PMLR, Atlanta, GA, USA, pp. 1139–1147, URL: https://proceedings.mlr.press/v28/sutskever13.html.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: IEEE International Conference on Computer Vision and Pattern Recognition. Boston, MA, USA, pp. 1–9. http://dx.doi.org/10.1109/CVPR.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: IEEE International Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, pp. 2818–2826. http://dx.doi.org/10.1109/cvpr.2016.308.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V., 2019. MnasNet: Platform-aware neural architecture search for mobile. In: IEEE International Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA, pp. 2815–2823. http://dx.doi.org/10.1109/cvpr.2019.00293.

Tan, M., Le, Q., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 97, PMLR, Long Beach, CA, USA, pp. 6105–6114, URL: https://proceedings.mlr.press/v97/tan19a.html.

Tan, M., Le, Q., 2021. EfficientNetV2: Smaller models and faster training. In: Meila, M., Zhang, T. (Eds.), International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 139, PMLR, pp. 10096–10106, URL: https://proceedings.mlr.press/v139/tan21a.html.

Thomas, A., Rider, M., Curtis, A., MacArthur, A., 2011. Automated lithology extraction from core photographs. First Break 29 (6), http://dx.doi.org/10.3997/1365-2397.29.6.51281.

Wanderley, J.F.C., Fisher, M.H., 2001. Multiscale color invariants based on the human visual system. IEEE Trans. Image Process. 10 (11), 1630–1638. http://dx.doi.org/10.1109/83.967391.

Ward's Science, 2023. Classic North American rock collection. Online. URL: https://www.wardsci.com/store/product/8869973/classic-north-american-rock-collection.

Wouwer, G.V.d., Scheunders, P., Dyck, D.V., 1999. Statistical texture characterization from discrete wavelet representations. IEEE Trans. Image Process. 8 (4), 592–598. http://dx.doi.org/10.1109/83.753747.

Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: IEEE International Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA, pp. 5987–5995. http://dx.doi.org/10.1109/cvpr.2017.634.

Zadeh, L.A., 1965. Fuzzy sets. Inf. Control 8 (3), 338–353. http://dx.doi.org/10.1016/s0019-9958(65)90241-x.

Zagoruyko, S., Komodakis, N., 2017. Wide residual networks, CoRR abs/1605.07146. pp. 1–15, arXiv:1605.07146.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., Smola, A., 2022. ResNeSt: Split-attention networks. In: IEEE International Conference on Computer Vision and Pattern Recognition Workshops. New Orleans, LA, USA, pp. 2735–2745. http://dx.doi.org/10.1109/cvprw56347.2022.00309.

Zheng, D., Zhong, H., Camps-Valls, G., Cao, Z., Ma, X., Mills, B., Hu, X., Hou, M., Ma, C., 2024. Explainable deep learning for automatic rock classification. Elsevier Comput. Geosci. 184, 105511. http://dx.doi.org/10.1016/j.cageo.2023.105511.