Automated Porosity Estimation using CT-scans of Extracted Core Data

Kurdistan Chawshin[1], *Carl Fredrik Berg* [1], *Damiano Varagnolo*[2], *and Olivier Lopez*[3]

[1]Department of Geoscience and Petroleum, NTNU. S. P. Andersens veg 15, 7031 Trondheim, Norway
[2]Department of Engineering Cybernetics, NTNU. O. S. Bragstads Plass 2D, 7034 Trondheim, Norway
[3]Equinor ASA. Arkitekt Ebbells veg 10, 7053 Ranheim, Norway

Abstract. Estimation of porosity at a millimeter scale would be an order of magnitude finer resolution than traditional logging techniques. This enables proper description of reservoirs with thin layers and fine scale heterogeneities. To achieve this, we propose an end-to-end convolutional neural network (CNN) regression model that automatically predicts continuous porosity at a millimeter scale resolution using two-dimensional whole core CT scan images. More specifically, a CNN regression model is trained to learn from routine core analysis (RCA) porosity measurements. To characterize the performance of such approach, we compare the performance of this model with two linear regression models trained to learn the relationship between the average attenuation and standard deviation of the same two-dimensional images and RCA porosity. Our investigations reveal that the linear models are outperformed by the CNN, indicating the capability of the CNN model in extracting textures that are important for porosity estimations. We compare the predicted porosity results against the total porosity logs calculated from the density log. The obtained results show that the predicted porosity values using the proposed CNN method are well correlated with the core plug measurements and the porosity log. More importantly, the proposed approach can provide accurate millimeter scale porosity estimations, while the total porosity log is averaged over an interval and thus do not show such fine scale variations. Thus, the proposed method can be employed to calibrate the porosity logs, thereby reducing the uncertainties associated with indirect calculations of the porosity from such logs.

Keywords: Whole core CT scan, Porosity, Convolutional Neural Network, Regression

1. Introduction

A comprehensive formation evaluation requires investigating different reservoir characteristics such as formation pressure, temperature, pore and grain size distribution, fluid saturations, permeability, and porosity. Out of these, porosity is one of the most important properties for reservoir modelling and reservoir characterization processes, as its value and variation within the reservoir determines the volume of reserves. Porosity is moreover strongly correlated to the reservoir quality with respect to the transport properties.

A common method for porosity estimation employs well logs such as density, neutron, and sonic logs, where porosity is indirectly computed from log responses by means of theoretical and empirical calculations. This method provides continuous porosity values along the well. The uncertainty in this type of porosity estimation is high since the logs are affected by other rock and fluid properties in addition to the porosity, e.g., lithology, type of fluids in the pore spaces, wellbore environment and type of drilling mud. Therefore, log-derived porosity interpretations need to be corrected and calibrated against core plug porosity measurements. Moreover, log-derived porosity interpretations overlook high resolution (i.e., below log resolution) porosity variations associated with fine-scale heterogeneities and thin layers.

Laboratory measurements using core plugs (typically 1 to 1.5 inch in diameter) and/or sidewall cores (1 inch in diameter) provide local porosity estimates that represent an interval of the cores extracted from the wells. Such analysis can produce acceptable reservoir porosity estimations for relatively homogenous pore systems. However, core plugs, normally sampled once per foot, might not provide representative data in heterogeneous reservoirs with millimeter scale porosity variations. More frequent sampling is necessary in this kind of reservoirs, which is laborious and cost consuming.

X-ray computerized tomography is a non-destructive technique that was initially used in medical science, and that has now been widely applied to investigate materials in the earth sciences, including petroleum geoscience [1], [2]. Whole core CT scanning, that provides two-dimensional and three-dimensional digital representations of extracted cores, has a long history in assisting geologists. Utilizing this technique leads to cores representations as grayscale images, where the gray-level values represent X-ray attenuation depending on the density and atomic number of the underlying materials. Core CT scans can provide valuable information on the texture and internal structure of reservoir rocks [3]. Recent developments in artificial intelligence techniques, combined with the fact that whole core CT scans are stored digitally, open new possibilities to enhance utilization of such data in rock characterization workflows. In this regard, CT scan images can be employed for automatic estimation of porosity at the millimeter scale, thereby enhancing their value in operational settings.

Artificial intelligence has been widely applied to predict transport properties such as porosity, permeability and water saturation using various well logs and/or a combination of core analysis measurements and well log data [4–20]. Ahmadi and Chen [4] compared different machine learning models for predicting porosity and permeability using the petrophysical logs and core analysis measurements. Aïfa et al. [6] proposed a Hybrid Neuro-Fuzzy model to predict porosity and permeability from well log data in the shaly sand reservoirs. Helle et al. [11] developed two back-propagation artificial neural networks (ANN) to predict porosity and permeability from density, gamma ray, neutron porosity, and sonic logs.

Moreover, several publications have evaluated the potential of deep neural networks in predicting transport properties (e.g., porosity and permeability) based on pore-scale micro-CT and Scanning Electron Microscopy (SEM) images [21–29]. Hebert et al. [24], inspired by Sudakov et al. [28], used an AutoEncoder to segment three-dimensional grayscale micro-CT images of Berea samples and to estimate porosity values using expert-derived segmented images as training ground truth. Bordignon et al. [23] proposed a methodology to estimate grain size and porosity distribution using synthetic idealized rocks (sphere packs) and Convolutional Neural Networks (CNN). Alqahtani et al. [21] employed deep learning to predict porosity, coordination number, and average pore size, where the ground truth values were computed using pore networks extracted from manually segmented images. More specifically, they used a CNN regression model to estimate the mentioned properties from two-dimensional grayscale micro-CT images. In the mentioned publications, machine learning and deep learning method were applied at the pore (micrometer) scale as an extension of Digital Rock Analysis (DRA).

In this study we propose a methodology to estimate porosity at the whole core (millimeter) scale using two-dimensional core CT slices. More specifically we employ an end-to-end CNN regression scheme to estimate porosity in a well using a training dataset that includes porosity measurements derived from routine core analysis. The trained model is then used to populate the whole well with porosity values. The obtained results are compared with a total porosity log computed based on the density log. We moreover compare the deep learning results with a simple linear regression (LR) model trained on average attenuations of the same two-dimensional image slices, and a multiple linear regression (MLR) model trained on both average attenuations and standard deviation of two-dimensional image slices. In fact, we discover that the CNN model outperforms the linear regression models, which indicates the optimized convolutional kernels extract the textures and patterns that are important for estimating porosity. Moreover, we quantitatively demonstrate that the proposed CNN model can be employed to provide accurate local porosity estimations.

The subsequent sections of this work are organized as follows: Section 2 presents a detailed description of the employed methodology, i.e., convolutional neural networks. Section 3 describes the image pre-processing steps, including the processes that have been applied to remove images with undesired image artefacts and perform image augmentation. Section 4 provides a brief description of the lithology of the studied formations together with the available data, including whole core CT scans and core analysis measurements. Section 5 describes the details of the model training and CNN hyperparameter selection processes. Section 6 summarizes and discusses the obtained training and prediction results. Finally, the last section provides some concluding comments, emphasizing the advantages of the proposed method for porosity estimation over other schemes.

2. Method

In this study we compare two approaches for automatic porosity estimation at the core scale. As shown in Fig. 1, the methodology starts with pre-processing of two-dimensional image slices, where images with undesired artefacts are flagged and removed. The pre-processing steps are explained in detail in Section 3.1.

After image pre-processing, two approaches are considered: In the first, the remaining pre-processed images are augmented to expand the training dataset, as CNN networks are proven to show better generalization capabilities with a higher number of augmented images. Then, the augmented images together with porosity values from routine core analysis are employed to train a CNN regression model. In the second approach, a simple (LR) model and a multiple linear regression (MLR) model are trained to predict porosity. The LR model is trained to learn the relationship between average attenuations of the two-dimensional images and RCA-derived porosity values. To count for distribution of the gray-level attenuations, we compute the standard deviation of the two-dimensional images. The computed standard deviation and average attenuations are used to train the MLR model to learn the measured RCA porosity. The reason for choosing linear models is that we expect a linear correlation between the average gray-level attenuations and porosity for a monomineralic rock type with one type of fluid [30, 31]. This comparison will reveal if the CNN model extracts pattens and features rather than just learning the average attenuation of the images. The training results of these approaches are compared, and finally the best performing model, i.e., the CNN, is used to predict porosity of unseen images and to populate the whole well with millimeter scale porosity values. The theory of convolutional neural networks will be summarized in subsequent sections.

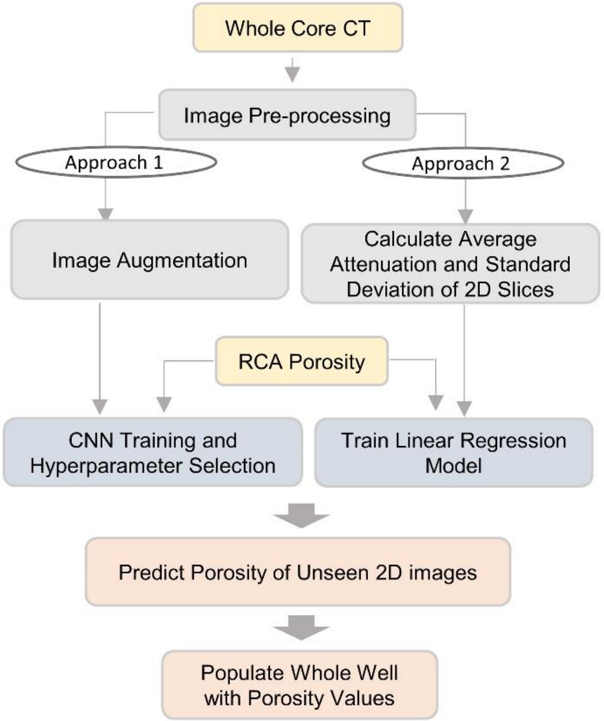
Whole Core CT
Image Pre-processing
Approach 1
Approach 2
Image Augmentation
Calculate Average Attenuation and Standard Deviation of 2D Slices
RCA Porosity
CNN Training and Hyperparameter Selection
Train Linear Regression Model
Predict Porosity of Unseen 2D images
Populate Whole Well with Porosity Values

Fig. 1. Summary of applied methodology for automatic porosity estimation of two-dimensional CT image slices.

2.1. Convolutional neural networks

Convolutional neural networks, initially proposed by LeCun [32], have been one of the most popular machine learning algorithms employed especially in classification, regression, and object detection tasks that involve images. CNNs consist of convolutional and pooling layers in addition to the fully connected layers of classical neural networks. The network in the convolutional section is locally connected, in contrast with fully connected layers where each neuron is fully connected to the neurons in the previous layers, and each connection has its own weight. The convolutional section in the CNN models consists of three main layers: convolutional, activation, and pooling. The sequence of these stacked layers can be repeated many times depending on the task at hand. The convolutional layers are responsible for feature extraction during training, where low level features are extracted in the first convolutional layer, while higher level features are extracted hierarchically as the output of a layer feeds into the next layer. The extracted features are then mapped into the output layer by fully connected layers [33]. The main elements of CNNs and their tasks are described below.

2.1.1. Convolutional layers

In the convolutional layers, a set of optimizable kernels are superposed on the two-dimensional or three-dimensional digital images represented as arrays of pixel numbers. Each kernel is convolved across the image using an element-wise multiplication between the kernel and the current area of the image that it covers (a.k.a. its *receptive field*). Then, these dot products are summed up and stored in the corresponding position of the output, i.e., the feature map. Once the convolutional operation is computed and stored for a specific position, the kernel is moved by a pre-defined offset, called stride. The convolution process is then repeated until the entire image has been traversed by the kernel. Since convolution is a linear operation, the output feature maps are passed through an activation function to introduce a nonlinearity. Currently the most common activation function to accomplish this is the rectified linear unit (ReLU). However, other activation functions, such as hyperbolic tangent or sigmoid functions, can also be employed. The convolution operation in the convolutional layers can be written as (modified from Anjos et al. [34]):

$$o_i = g(o_{i-1} * W_i), \qquad (1)$$

where o_i is the output of the i^{th} layer, o_{i-1} is the output of the previous layer, $*$ is the convolution operation, W_i is the kernel weights of the i^{th} layer, and g is the nonlinear activation function.

2.1.2. Pooling layers

The feature map outputs of the convolutional layers record the exact position of the features; that said, minor spatial changes in the input image result in a different feature map. Therefore, after applying the activation function, a pooling layer is added with two purposes. First, to make the extracted feature maps invariant to local translations and spatial variations in the input image. Second, to reduce the spatial dimension of the feature maps, thereby reducing the computational time associated with model training. Pooling is analogous to the convolutional operation, where a window is sliding over the input image. However, it executes a selection of elements inside the pooling window without trainable parameters. The most common pooling operations are maximum pooling and average pooling. Maximum pooling picks the maximum value inside the window and discards the others, while average pooling computes the average of values inside the pooling window. The down-sampled feature map output of the pooling layer is called the pooled feature map, and it is expected to contain the most important structural features.

2.1.3. Fully connected layers

The pooled feature maps of the convolutional section are then flattened and used as input to the fully connected layers, whose purpose is to map the extracted features into the output layer. As opposed to the convolutional layers, here the input nodes are fully connected to the output nodes by learnable weights. Nonlinearity in the fully connected layers can be introduced by adding an activation function such as ReLU so that mathematically these layers can be explained as (modified from Anjos et al. [34]):

$$z_i = g(z_{i-1} W_i + b_i), \qquad (2)$$

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

One of the limitations of the CNN architecture described above is that it can easily overfit to the training data, and thus diminish the generalization capabilities of the model. To overcome this limitation one may employ some regularization technique, such as dropout, where randomly selected neurons are temporarily removed together with their incoming and outgoing connections, and not used in the back-propagation phase [35, 36]. Dropout is performed to optimize the bias-variance tradeoff, i.e., balancing out the bias and variance of the estimates towards better performance. Batch normalization is another common regularization technique, where the output of a convolutional layer is normalized before being used in the next one. Batch normalization is empirically known to make training less sensitive to the initialization point, plus speeding up the network training [37].

3. Data pre-processing

The provided CT scan data for individual cores are stored in 16bit unsigned DICOM format [38]. We stack the DICOM images of individual core scans together and store them as three-dimensional raw images using the *ImageJ* software [39]. Once the three-dimensional raw images are created, the next step is to remove border effects by cropping the image into rectangular crops. We used a centered crop of size 256×256 pixels, as shown in Fig. 2. Then a global minimum and maximum intensity value is assigned to the rectangular crops of the individual cores. These global intensity values are selected by observing the three-dimensional histograms of rectangular crops of the entire considered core. Further, the intensity adjusted images are encoded in 8bit format, i.e., $0 - 255$ grayscale values.

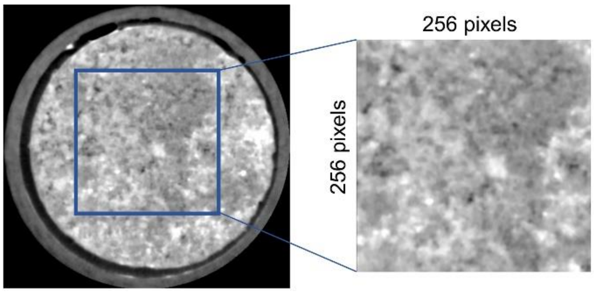

Fig. 2. As a pre-processing step, the original DICOM images are cropped into crops of size 256×256 pixels (blue square). This is done to remove border effects.

3.1. Removing undesired artefacts

Some of the studied CT images contain non-core regions and undesired artefacts. One type of such artefacts is missing pixels associated with poor core recovery, induced fractures, or rush plugs. Images with missing pixels show low gray-level attenuation values (Fig. 3-a). Other types of undesired features are related to the intervals with drilling mud invasion, high-density minerals (e.g., pyrite and siderite) cementation (Fig. 3-b), and core barrel couplings (Fig. 3-c).

The images with artefacts can negatively affect the performance of the considered machine learning algorithms, since they contain features not related to the actual phenomena being modeled. Thus, we implemented specific type-dependent algorithms that aim at excluding images with such artefacts. More precisely, to remove missing intervals we calculate the average attenuation μ_c at the center of the image using a centered square covering 40% of the total number of pixels. If the computed average attenuation is less than a pre-defined cutoff C_m, the image is flagged and removed.

$$f_m = \begin{cases} 1, & \mu_c < C_m \\ 0, & else \end{cases} \tag{3}$$

where f_m is the flag for missing interval. The image is removed if f_m is equal to 1. The cutoff C_m is defined by the user based on observing the average attenuation of images with missing pixel values. In this study, we used cutoff $C_m = 50$.Image intervals with drilling mud invasion and high-density material show high gray-level attenuation readings, and they appear very bright compared to the other intervals. To identify these intervals, we compute the average attenuation μ of the whole two-dimensional image, and if the average attenuation is greater than a pre-defined cutoff C_h, the image is flagged for removal:

$$f_h = \begin{cases} 1, & \mu > C_h \\ 0, & else \end{cases} \tag{4}$$

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

The cutoff C_h is defined by the user based on observing the average attenuation of images with drilling mud invasion and high-density minerals. In this study, we used cutoff $C_h = 170$.

Images from intervals with core barrel couplings show low gray-level attenuation readings in the middle, while the image edges appear brighter with higher attenuation readings (Fig. 3-c). To detect these intervals, the difference in average attenuation of the center and edges of the image slices is calculated. As above, the center average attenuation μ_c is computed considering 40% of the total number of pixels using a centered square. To represent edge average attenuation μ_e, the outer 5% of the total number of pixels along the edges are considered. If the difference between center average attenuation and edge average attenuation is greater than a pre-defined cutoff value C_b, the image interval is flagged and removed:

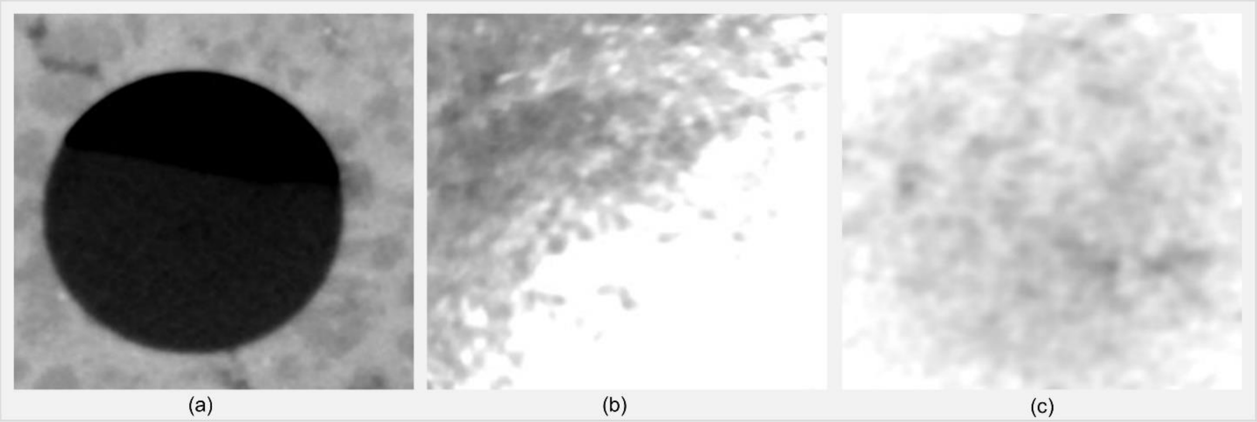

Fig. 3. Two-dimensional image slices with undesired artefacts: (a) missing pixels, (b) high density material, (c) core barrel coupling

$$f_b = \begin{cases} 1, & \mu_e - \mu_c > C_b \\ 0, & else \end{cases} \tag{5}$$

where f_b is the core barrel coupling flag. The image is removed if f_b is equal to 1. It is worth to mention that the employed thresholds were computed using the global distribution of the minimum, mean, and maximum intensity values observed in the data set. Note that in the first approach (Fig. 1), the images are normalized before being used as input for the CNN training (i.e., images are divided by 255). Moreover, to reduce the computational time associated with model training, the remaining two-dimensional image slices are coarsened by a factor of four, so the final image size as used for input to the CNN is 64×64 pixels.

3.2. Image augmentation

Deep learning networks notoriously require large amounts of training data to achieve good performance. Aware of this, we augmented the training images used for the CNN through image augmentation, i.e., by applying different kinds of modifications (e.g., shifting, flipping, random rotation, and shearing) to the original images. In this way the model can learn from more examples during training phase, and at the same time make the model less sensitive to spatial translations. In this study we specifically considered rotation and horizontal flips of the two-dimensional images. In the case of horizontal flip, we employed the "ImageDataGenerator" class, a publicly available code that can be used for image augmentation purposes on the fly, in Python using the Keras API [40]. The "ImageDataGenerator" class rotates the images randomly within a range of user-defined angles, where in case of squared images it is very likely that for some specific rotation angles the pixels will fall out of the image frame leaving some areas of the image with no pixels. There are several interpolation techniques, such as nearest neighbor, that can be used for those areas, but they can amend the key features and create dissimilar features that are then counterproductive for the training process. To avoid this problem, the images were rotated outside Keras, while the horizontal flipping was performed on the fly using the "ImageDataGenerator" class. Here we thus eventually considered rotation of the original images by 90°, $180^{\circ}$and 270°.

The dataset

4.1. Lithology description of the studied well

This study utilizes the whole core CT scan images of a well on the Norwegian continental shelf. The well penetrates four main formations, denoted as Fm. (Formation) 1, Fm.2, Fm.3, and Fm.4 in Fig 4. Formation 1 consists of very fine-grained argillaceous sandstones and cemented sandstones; Formation 2 consists of successive layers of mudstones and fine-grained sandstones; Formation 3 consists of granule rich medium-grained sandstones and spiculites (a biogenic rock composed of sponge silica spicules); Formation 4 comprises mud and calcite rich marlstones. The overall lithology in this well is divided into 20 lithofacies by manual core descriptions [41]. A brief description of different lithofacies together with associated abbreviations is provided in Table 1.

4.2. Whole core CT scan images

As mentioned previously, in this study we employ two-dimensional core CT images. The original images
are stored as 16bit unsigned DICOM images slices with a vertical resolution of approximately 0.45 millimeters. The images are pre-processed, i.e., cropped and discarded if presenting undesired artefacts as explained in section 3. The remaining images are coarsened to 64×64 pixels and normalized before being used for further analysis.

4.3. Routine core analysis data

Porosity measurements of 388 core plugs from routine core analysis (RCA) are considered as ground truth for the considered convolutional neural network and linear regression models. The RCA porosity-permeability cross-plot is shown in Fig. 5, where different colors represent different lithofacies observed through core descriptions.

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

Table 1. A brief description of the observed lithofacies and their associated abbreviations derived from manual core descriptions.

According to Fig. 5, the studied well contains a whole range of porosity values, from approximately 0.03 for well cemented sandstones (WCemMSS) up to 0.40 for cross stratified sandstones (CrossFineSS). Permeability values range from 0.01 mD for mudstones up to 50 Darcy for granule rich medium-grained sandstone samples (GraMSSDispC).

5. Training

As mentioned previously, convolutional neural networks and linear regression models were employed in two separate approaches to estimate porosity at the whole core scale. This section presents the details of the training processes for both approaches.

5.1. Labeling and train-test splitting

In this study the RCA-derived core plug porosity measurements were used as the target porosity values for individual two-dimensional images. Porosity is often measured on core plugs with typical lengths of 1 to 2 inches [42], which corresponds to approximately 55 to 110 two-dimensional CT image slices with vertical resolution of 0.45 millimeters. In our approach we assign the same porosity labels to 19 successive two-dimensional images (approximately 8.5 mm vertical length) at depth intervals corresponding to the core plug depths. In other words, individual two-dimensional images are used as input to train the CNN model, where 19 successive images at each depth interval are labeled with the same porosity values coming from core plug measurements. Note that the number of images in specific intervals might be less than 19 since images with artefacts are removed by image pre-processing.

For regression modeling we use the same labeling strategy. However, in case of linear regression, the average attenuation and standard deviation of each two-dimensional image is computed. The simple LRmodel is trained to learn the relationship between average attenuations (as independent variables) and core analysis porosity measurements (as dependent variables), while the MLR model is trained to learn RCA measured porosity from both average attenuation and standard deviation.

Generally, a machine learning model is trained on a set of samples, i.e., training set. In a supervised learning environment, the training dataset consists of pairs of input and corresponding output values that are used as the training target. The output predictions during training are compared with the actual target values, and the model parameters are updated to minimize the error between actual target and predicted values. Once the model is fitted to the training set, its statistical performance is evaluated on a set of unseen instances, i.e., test set. In this manner, the test set provides an unbiased evaluation of the trained model.

The labeling strategy explained above resulted in 6863 labeled images (about 3% of the total number of images in the studied well). To cover the distribution of the measured porosity values, we chose ~90% of the images as the training set and the remaining ~10% as the test set. The train and test sets were separated manually to ensure that the test set was representative for the porosity distribution seen in the well. The test set is a continuous subset of the well, therefore we do not have images from the same plug in the train and test sets. Since the lithofacies classes are varying gradually and slowly, a

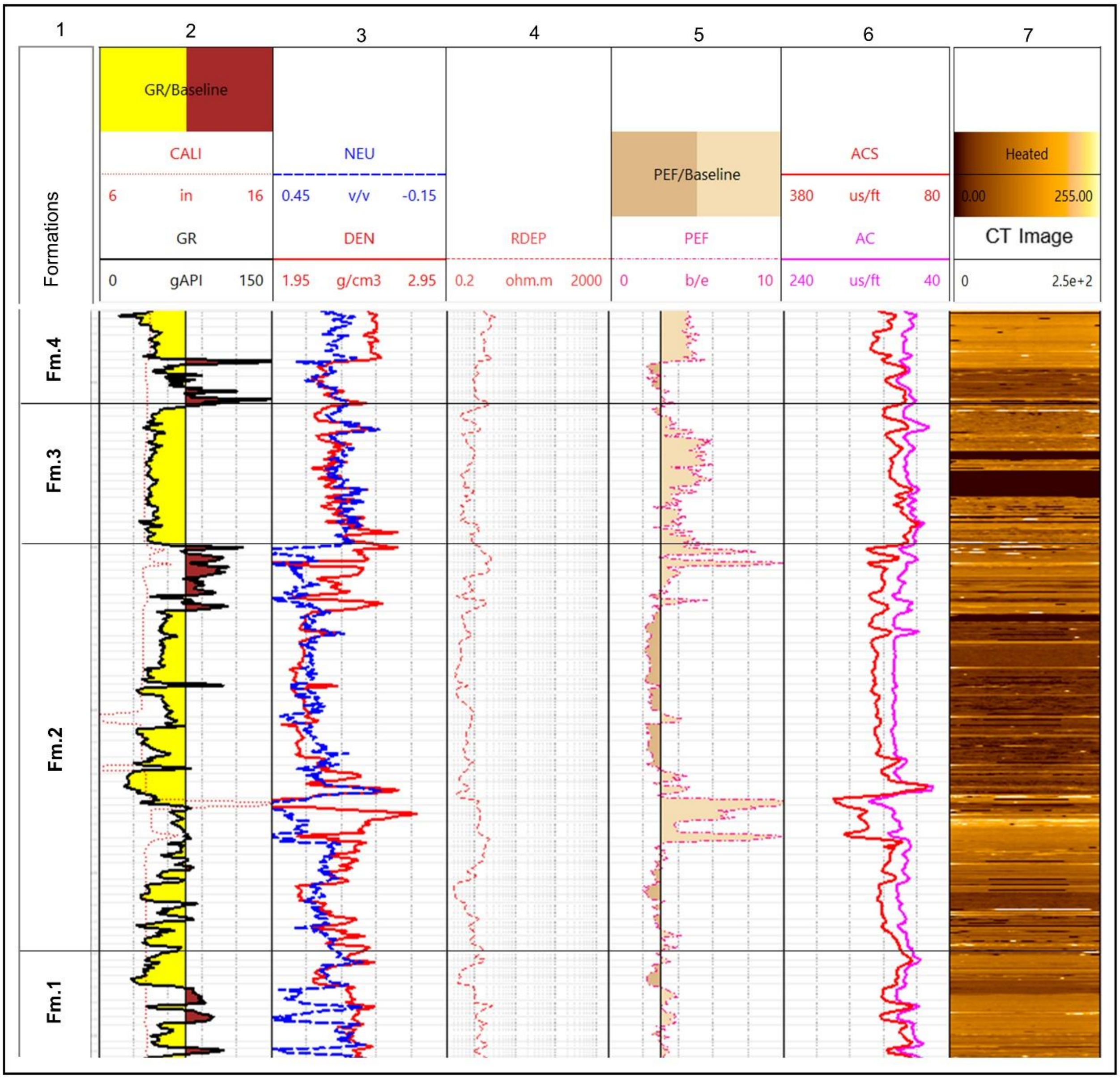

Fig. 4. Well log data and two-dimensional cross section of the core CT image showing 142 meters of the studied well. Log tracks from left to right: track 1: Formations, track 2: Caliper (CALI) and Gamma ray (GR), track 3: Density (DEN) and Neutron (NEU), track 4: Deep resistivity (RDEP), track 5: Photoelectric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

random spilt would result in similar images in the training and test sets, and overestimation of the performance of the model. Moreover, 20% of the training set was employed as the validation set to provide an unbiased evaluation of the model while tuning the kernels and the weights in the network. As mentioned before, the images used for CNN training were further augmented (rotated by three angles of 90°, 180°and 270°) to expand the training dataset and to obtain a more robust model that is invariant to the image modifications. The image augmentation process resulted in 19764 training images.

5.2. CNN training and hyperparameter selection

Commonly, CNN training is performed by forward and back-propagation processes, where the kernels in the convolutional layers and weights in the fully connected layers are updated iteratively to minimize the error (cost function) between the actual and predicted values. Mean squared error (MSE) and mean absolute error (MAE) are commonly used as cost functions in CNN regression problems, and calculated as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \qquad (6)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \qquad (7)$$

where N is the total number of samples, while y_i and $\hat{y}_i$ are the actual and predicted values of sample i.

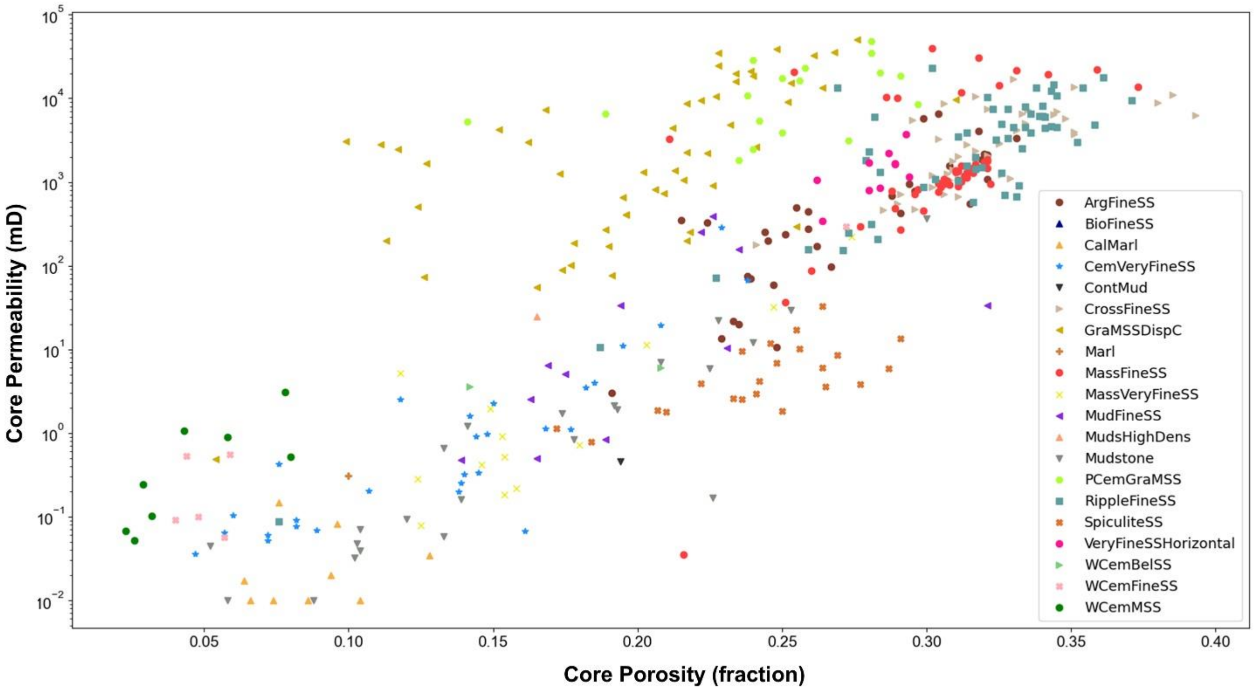

Fig. 5. Porosity-permeability cross-plot for available core analysis measurements of the studied well. Different colors represent various lithofacies observed through core descriptions.

The cost function is minimized using a gradient descent optimization algorithm. More precisely, the partial derivative of the cost function with respect to each learnable parameter is calculated and those parameters are updated using [33]:

$$\omega^{+} = \omega - \alpha \frac{dE}{d\omega}, \qquad (8)$$

where ω refers to each learnable parameter (i.e., weights and kernels), α is the learning rate, and E is the cost function. In this study we considered both types of the above cost functions. However, based on our preliminary experiments the optimization of the MAE leads to a model with better performance on the validation set than a model optimized with MSE. Therefore, we used MAE as the cost function to be minimized during model training.

The learning rate determines how fast the learnable parameter should move along the direction of the gradient, and its value can significantly affect the convergence of the cost function optimization process. There are different types of optimizers and descent methods to minimize the cost function. Here, we specifically used the Adam optimizer [43] and mini-batch gradient descent to search for optimum weights and kernel parameters. Mini-batch gradient descent is the most common variation of gradient descent in deep learning [44], where the training dataset is split into small batches, and the error is calculated per batch before updating the learnable parameters. In our study the final optimal approach was to consider a batch size of 32 images. The optimization procedure is carried out using a pre-defined number of epochs, where an epoch is a period in which all the training samples have been presented at least once to the network. The number of training epochs has high impact on the performance of the CNN model too, since a too large or too small number of epochs can lead to over- and underfitting, respectively. To overcome this issue, in our study we used an early stopping regularization technique, i.e., make the model training stop once the model performance on the validation dataset starts to degrade after a certain number of epochs. Here, a total number of 300 epochs was assigned for the training process.

In addition to the learnable parameters there exist other types of parameters, so called hyperparameters, that are not learnable during the training process – for example the number of convolutional layers, the number of kernels in the convolutional layers, the kernel size, the learning rate, and similar factors.

The values of such hyperparameters are typically initially assigned by the user using domain expertise considerations and are kept constant during the training phase. Since the choice of the hyperparameters highly affects the performance of CNN networks, such hyperparameters may be adjusted through a process referred to as hyperparameter tuning. In this study we performed such a tuning on the number of convolutional layers, the number of kernels in the convolutional layers, kernel size, learning rate, number of neurons in the fully connected layer and dropout rate.

The proposed CNN model was developed in Keras with the Tensorflow backend. To automatically tune the considered hyperparameters, we employed the Keras tuner library [45], where we define a search space for each hyperparameter together with a proper tuner. The tuner automates the tuning process by evaluating a certain number of configurations with different hyperparameter combinations. The considered hyperparameters and associated search spaces are presented in Table 2. There are different types of tuners such as RandomSearch,

CNN hyperparameters	Parameter space
Number of convolutional layers	(2, 3, **4**, 5)
Number of convolutional kernels	(16, 48, 80, **112**, 144, **176**, 208, **240**)
Kernel size	(3, **5**)
number of neurons in the fully connected layer	(32, 64, 96, 128, **160**, 192, 224)
Learning rate	(0.01, 0.001, **0.0001**)
Dropout rate	(0, **0.2**, 0.4, 0.6)

Table 2: The CNN hyperparameters and their associated search space considered during hyperparameter tuning to optimize the model architecture. The final optimal values are shown in bold. Note that three numbers are bold for convolutional kernels since two convolutional layers have 176 kernels each, while the other two have 112 and 240 kernels each (Fig. 6).

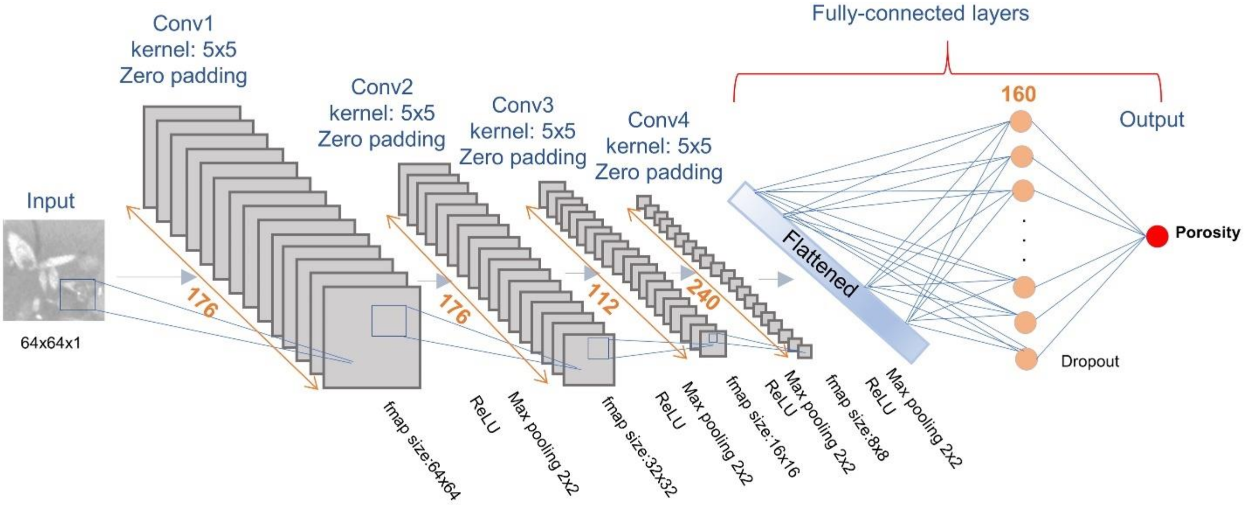

Fig. 6: The proposed CNN architecture for porosity estimation using two-dimensional CT images.

Hyperband and BayesianOptimization. More detailed information on the differences among these approaches can be found in [46–49]. In this study we used the Hyperband algorithm [48], which is a relatively new method, and that uses adaptive resource allocation and an early stopping rule to quickly converge to a high-performance model. More precisely, a large number of random configurations are run for a specific number of epochs (i.e., one or two) per configuration. Then the top-performing model configurations based on the previous results are trained for more epochs. Finally, the best model configuration trained to the assigned maximum number of epochs is returned by the algorithm. The proposed CNN architecture, optimized using the above hyperparameter selection process, is indicated by bold numbers in Table. 2, while the corresponding CNN is shown in Fig. 6. The CNN architecture is then described in detail in the next subsection.

5.2.1. CNN architecture

The optimal CNN architecture is presented in Fig.6. The input images are two-dimensional grayscale images of size 64×64, with one channel represented by two-dimensional arrays. There are four convolutional layers, denoted as “Conv1”, “Conv2”, “Conv3” and “Conv4” in Fig. 6, with 176, 176, 112 and 240 convolutional kernels, respectively. From the hyperparameter tuning the optimal size of convolutional kernels is 5×5.

To preserve the original image size, zero padding is applied in the convolutional layers, i.e., a layer of pixels with values of zero are added around the edges of the images. Here, the convolution operation is applied using a stride of 1, i.e., the convolutional kernel is moved by 1 pixel while performing convolution operations.

To introduce nonlinearity, a ReLU activation function is applied to the feature map outputs of each convolutional layer. Then, the feature maps are sent to the next layer, the pooling layer, where they are downsampled using a pooling window size of 2×2 and a stride of 1.

The pooled feature maps of the convolutional section are flattened into a one-dimensional vector that is fed into the fully connected layer to be mapped to the output layer. The proposed architecture consists of one hidden layer with 160 neurons optimized by the hyperparameter tuning process explained above. Moreover, a ReLU activation function followed by a dropout layer is applied to the hidden layer. As explained in subsection 2.1.3., the dropout layer is applied as a regularization technique to prevent overfitting in the proposed network. Here, a dropout rate of 0.2, optimized during hyperparameter tuning, was applied in the proposed network, meaning that one in 5 of the neurons (in the hidden layer) together with their incoming and outgoing connections were randomly ignored from each update iteration.

The output layer contains a single neuron since we are dealing with a regression problem, where a single porosity value is predicted for each input image.

5.3. Linear regression training

As indicated before, two linear regression models (LR and MLR) were trained to predict porosity directly from the average attenuation and standard deviation values calculated for the individual two-dimensional image slices. The purpose of such a step is to enable a subsequent analysis revealing if the CNN model extracts relevant features rather than just learning the average attenuation and standard deviation of the images. Fig. 7 then shows the average attenuation versus measured porosity values for the training dataset, where different colors correspond to various lithofacies classes in the study well. Note that the same porosity values are assigned to 19 successive image slices. We can see that the number of images associated to some porosity values are less than 19 since some of the images with artefacts are removed by image pre-processing. This figure shows that porosity is negatively correlated with average attenuation. However, some of the datapoints behave differently and appear as outliers (see the red and blue ellipsoids in Fig. 7). Based on our investigations the datapoints within the red ellipsoid are associated with images that partly contain missing pixels, and this indicates that these images were not completely removed during the image pre-processing step. The missing pixels are characterized by lower gray-level attenuation values, resulting in lower average attenuation in these images compared to the other images with similar measured porosity values. The datapoints within the blue ellipsoid, instead, are associated with images containing high density material with higher average attenuation values compared to the images with similar measured porosity values. Since the linear regression training process can be affected by outliers, we included a step of detecting and removing these outliers prior to model training. To detect the outliers we employed the Isolation Forest (iForest) algorithm initially proposed by Fei Tony Liu et al. [50, 51]. The iForest algorithm is a model that is based on unsupervised learning and that works using the principle of isolating anomalies. Similar to the Random Forest, the iForest method randomly splits the datapoints by building an ensemble of trees (called iTrees), where the goal here is to isolate the anomalous datapoints. Based on this algorithm, there is a tendency for anomalous instances to be isolated easier compared to the normal instances, i.e., anomalies are the datapoints with short average path lengths on the iTrees.

As expected, the datapoints related to the images with partly missing pixels and high-density material (red and blue ellipsoids in Fig. 7) were detected as outliers by the iForest algorithm. These outliers are removed from the training set before fitting the linear regression models. The fitted LR model is shown in Fig. 7.

6. Results and Discussions

We now present the obtained training results and compare the prediction capabilities of the CNN and linear regression approaches presented above.

6.1. Training results

The CNN regression model was trained to predict RCA measured porosity values using two-dimensional whole core CT image slices. The linear regression models were instead trained to predict RCA measured porosity based on the average attenuation and standard deviation values computed on the same two-dimensional CT image slices. Note that the LR model was trained only on the average attenuations, while the MLR model was trained on both average attenuations and standard deviations with the purpose of counting for variation from the mean average attenuation. To evaluate the training results, we have shown a quantitative description of the proposed models in Table 3, where we can clearly see that the CNN outperforms the linear regression models. Moreover, we can see better performance in the linear regression models by including the standard deviation as one of the independent variables. The measured porosity is plotted versus the predicted porosity for the CNN and MLR models in the first row of Fig. 8. In addition, the corresponding residual plots are shown in the second row. We observe that the measured and CNN predicted porosity values are in good agreement, while there are bigger deviations between the measured and MLR predicted values. Moreover, the residual errors for the CNN model are randomly distributed around 0, while the residual errors for the MLR model exhibit patterns with various degrees of scattering for various fitted porosity values. These results indicate that average attenuation and gray-level standard deviation do not fully explain the porosity variations. In other words, the pixel intensity variations that are captured by the CNN model translate into a learning of the relationship that exist between these variations (extracted in the form of

Models	r^2	MAE
CNN	0.997	0.005
LR	0.768	0.027
MLR	0.781	0.026

Table 3: Coefficient of determination (r^2) and mean absolute error (MAE) of the training set for the CNN and LR models.

features) and the porosity values. Having highlighted this important result, in the remainder of the paper we continue with evaluating only the performance of MLR and CNN models in predicting porosity on the unseen images.

6.2. CNN evaluation

As mentioned, to obtain an unbiased evaluation of the CNN model, 20% of the training set was used as validation set during training and hyperparameter tuning process. In other words, once the model is trained at the end of each epoch, its generalization capability is evaluated on the validation set using the mean absolute error as the training metric. Fig. 9 shows how the error is minimized with increasing the number of epochs in both the training and the validation sets. Note that despite assigning 300 epochs, the training process stops after 221 iterations due to the early stopping regularization technique dictating so. We can see that the model shows high generalization capabilities on the validation dataset.

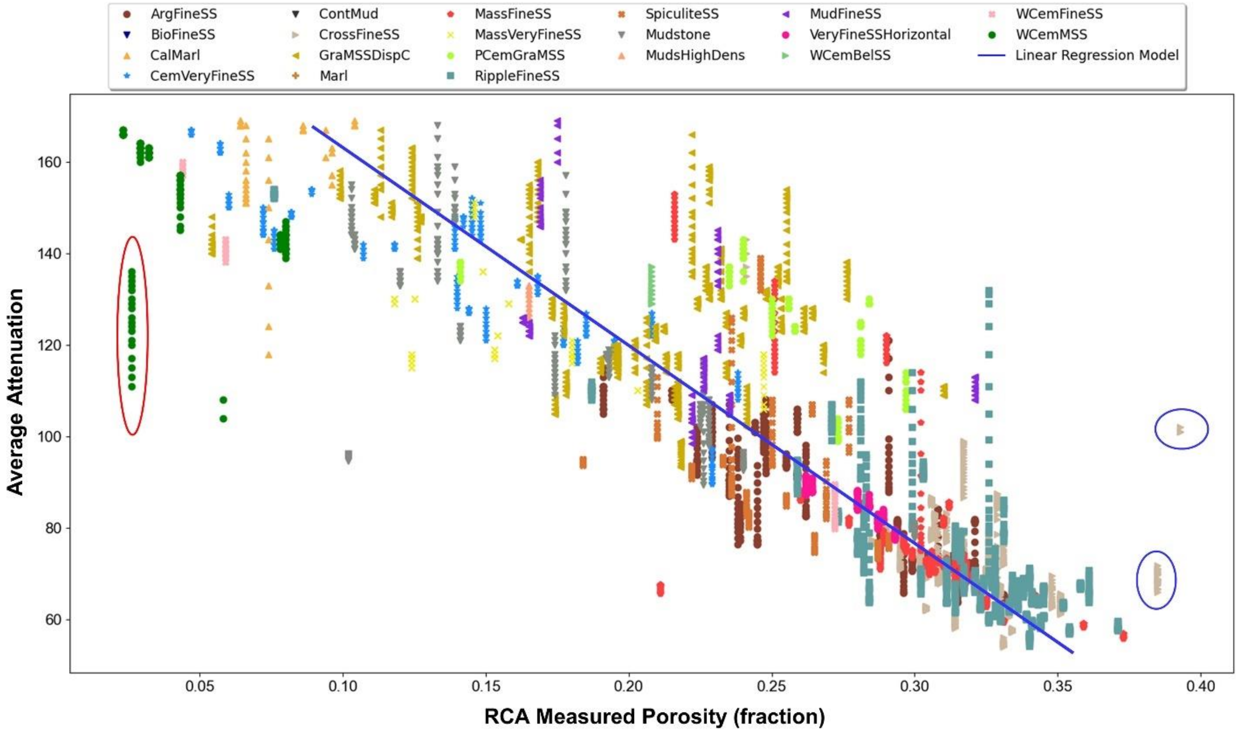

Fig. 7: Average attenuation of two-dimensional image slices versus RCA-derived porosity measurements. Different colors correspond to different lithofacies in the studied well. The datapoints within the red and blue ellipsoids are the outliers detected by iForest algorithm. The LR is trained on the datapoints excluding the outliers.

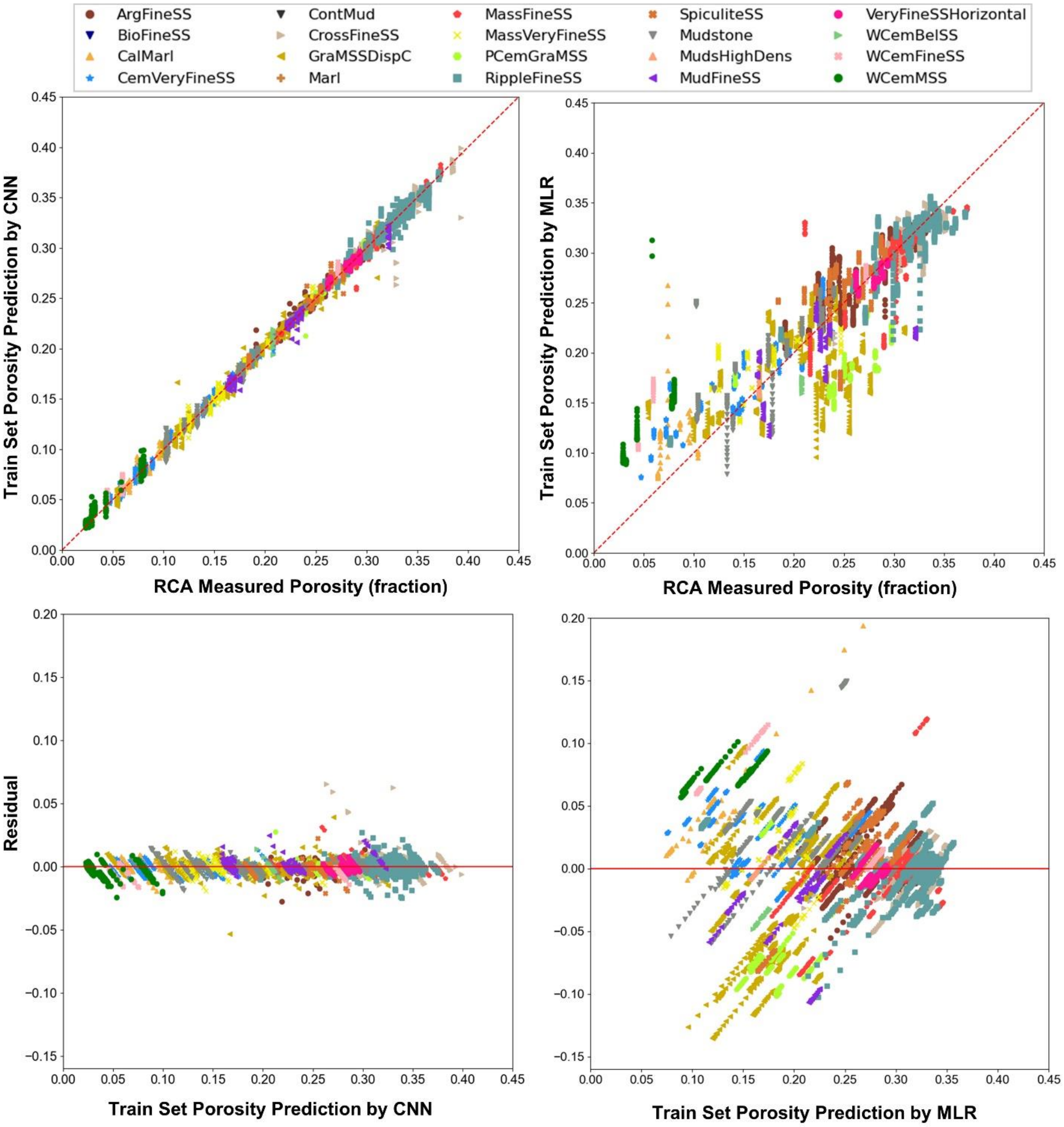

Fig. 8: Porosity prediction of the training dataset. In the first row, the predicted porosity is plotted versus the actual measured porosity for the CNN (left) and the multiple linear regression (right) model. The red dashed line represents the 1:1 line. The corresponding residual error plots are shown in the second row.

6.3. Porosity prediction

The proposed CNN and MLR models were further used to predict porosity on a set of unseen images, i.e., the test set, from the same well. Like the training set, 19 successive images at each depth interval were labeled with the same RCA derived porosity values. Note that in the case of MLR model, the calculated average attenuation and standard deviations of test set images were used to predict RCA porosity, while in case of CNN the image slices were directly used to predict porosity. The CNN predicted results are plotted versus the actual measured values in Fig. 10 (left). Here we show the CNN predicted porosity values with error bars, where the markers signify the mean predicted porosity of 19 successive images, while the error bars show the standard deviations, i.e., the variability of the predicted porosity for the 19 images. The mean predicted porosity and the actual measured porosity for the CNN and MLR models (Fig. 10) show an r^2 of 0.81 and 0.69, respectively. Here, we can see that the CNN predicted mean porosity values are closer to the line of equality (1:1 line) and the error bars for most of the samples cross the line of equality), showing that the correlation is balanced. These results again indicate higher performance of the CNN model in predicting porosity on the unseen images.

However, there are a few samples that occur relatively far below the equality line, i.e., samples belonging to the

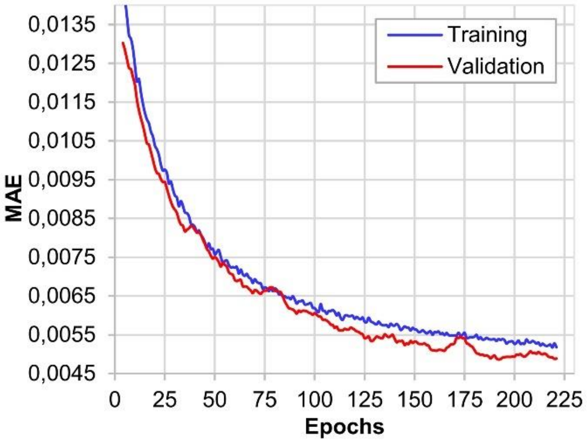

Fig. *9*: The CNN model performance during training using mean squared error as the evaluation metric. The blue and red lines depict the training and validation errors, respectively.

poorly cemented granule rich medium-grained sandstone (PCemGraMSS) and muddy fine-grained sandstone (MudFineSS) classes. These are shown within the red dashed ellipsoid in Fig. 10.

Image examples of these classes are presented in Fig. 11. One can clearly see that the images in the first row are characterized by high drilling mud invasion, where mud appears with very bright to white gray-level attenuation values. Moreover, the images in the second and third rows are darker in the middle and brighter at the edges. As mentioned before, this type of artefacts is characteristic of the image intervals with core barrel couplings. Apparently, these images are not completely removed during pre-processing, and the mentioned artefacts are probably the reason for CNN model deficiency in porosity prediction associated with these images. Here the model is predicting lower porosity values compared to the actual values. This is expected given the presence of pixels with high gray-level attenuations.

In addition, it is worth to mention that the proportion of different lithofacies classes is varying in the training set (as shown in Table 4), meaning that the CNN model is trained on fewer image examples for classes in minority than the more frequent ones. Given such an imbalanced training dataset, the model is expected to show lower prediction performance on the minority classes. This is clearly confirmed in Table 4, where the average prediction error for lithofacies classes in minority is higher than the one in more frequent classes. Note that high average prediction error for the PCemGraMSS class is associated with image artefacts (as explained above).

Overall, the proposed CNN model shows satisfying prediction performance on the unseen images. Therefore, we applied this model to predict porosity throughout the whole studied well, thereby populating the whole well with millimeter scale porosity values. Note that this interpolation step is performed using a model that was trained and evaluated on approximately 3% of the available images. The obtained results are presented in Fig. 12, where we can see a clear correlation between the CNN predicted porosity (light blue curve) and the RCA porosity (red circles). The available total porosity log is also plotted on the same figure, i.e., the black curve. The total porosity (φ) log is computed using density porosity equation:

Lithofacies Classes	Proportion in Training Set	Average Prediction Error
Marl	0.003	0.060
CalMarl	0.009	NA
SpiculiteSS	0.050	0.023
Mudstone	0.056	0.026
WcemBelSS	0.003	0.065
GraMSSDispC	0.149	0.009
PCemGraMSS	0.040	0.120
WCemMSS	0.021	NA
MudsHighDens	0.003	NA
ArgFineSS	0.091	0.003
RippleFineSS	0.149	0.002
MassFineSS	0.126	0.008
CrossFineSS	0.125	0.003
MudFineSS	0.025	0.028
BioFineSS	0.003	NA
WcemFineSS	0.009	0.020
ContMud	0.003	NA
MassVeryFineSS	0.034	0.006
CemVeryFineSS	0.081	NA
VeryFineSSHorizontal	0.022	0.005

Table 4: The proportion of various lithofacies classes in the training set and average prediction error calculated on the test samples of each class. Note that some of the classes are missing in the test set, so the average prediction error for those classes is not applicable.

$$\varphi = \frac{\rho_{ma} - \rho_b}{\rho_{ma} - \rho_{fl}}, \qquad (9)$$

where ρ_{ma} indicates the matrix density index obtained from core measurements, ρ_b is the bulk density index derived from the density log, and ρ_{fl} is the measured fluid density. To compare the predicted results with the total porosity log at the same scale, the predicted porosity was coarsened to match the resolution of the porosity log.

For this purpose, we calculated a moving average of the CNN predicted porosity using a window size of 30 cm. The coarsened porosity curve is shown by green color in Fig. 12, where one can see that it is well correlated with the total porosity log. A 1.5 meter interval of the well is zoomed in and shown in Fig 12. Here we can see the millimeter scale porosity variations predicted by the CNN model, while the other two curves are averaged and do not show finer scale variations. In fact, the predicted porosity is more accurate and more in line with the core porosity measurements compared to the total porosity log. Note that a bulk depth shift has been applied on the CNN images and RCA porosity measurements to match the depth of the total porosity log. As this is a bulk depth shift, there might be local depth-match issues.

The proposed method can thus be employed to provide accurate local porosity estimations and the obtained results can be used to calibrate porosity logs. Moreover, it can help in identifying the appropriate core plug locations.

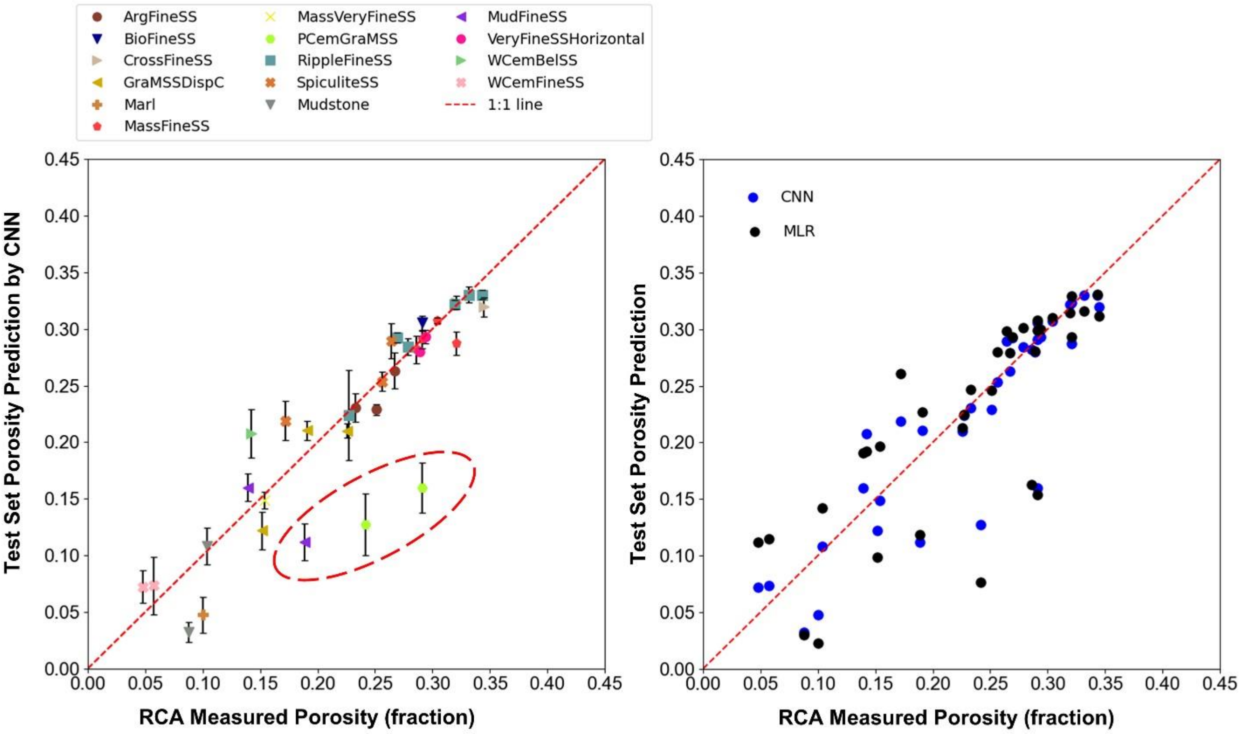

Fig. 0: CNN Porosity prediction of the unseen test images plotted versus the actual measured porosity (left). The CNN model shows an r^2 of 0.81, where high deviations occur in images with drilling mud invasion and core barrel coupling artefacts. Mean porosity prediction of the unseen test images plotted versus actual measured porosity for both the CNN and MLR models (right). The MLR model shows an r^2 of 0.69.

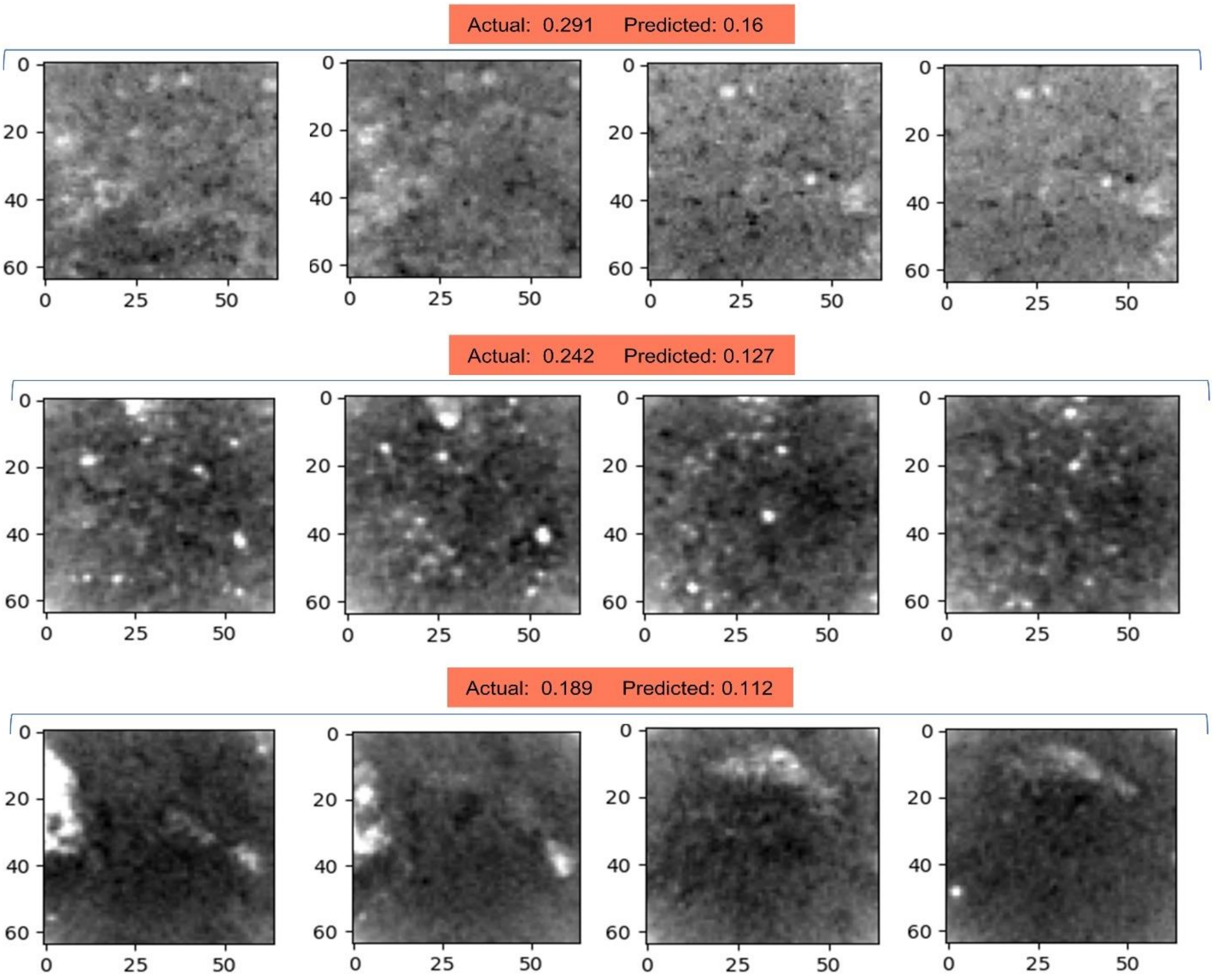
Actual: 0.291 Predicted: 0.16
Actual: 0.242 Predicted: 0.127
Actual: 0.189 Predicted: 0.112
0
20
40
60
0
25
50

Fig. 1: Examples of the two-dimensional images from the test set, where there are bigger deviations between the actual measurements and the predicted porosity values. Clearly, the image artefacts due to mud invasion (first row) and core barrel couplings (second and third row) are the reason for CNN model deficiency.

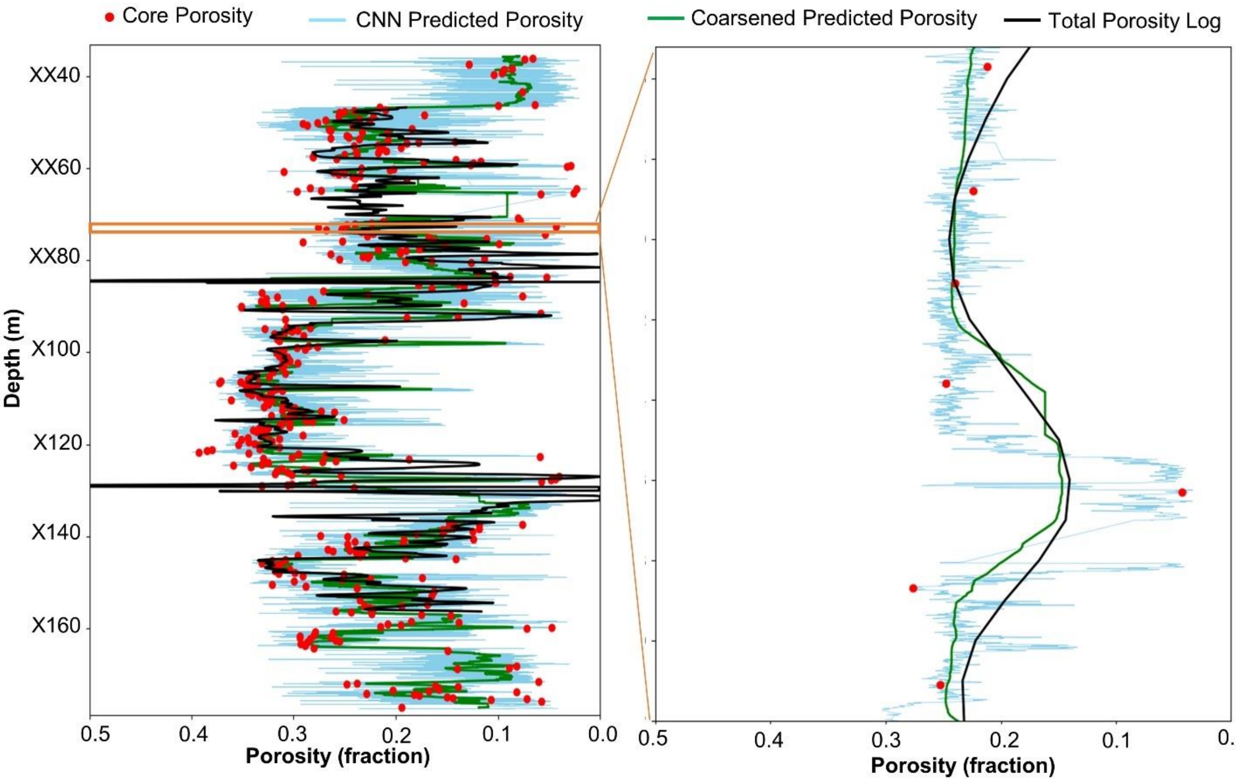

Fig. 2: Populating the whole well with porosity values. The CNN predicted porosity is in line with the RCA measured porosity. A 1.5 meter interval is zoomed in and shown in the plot to the right.

6.4. Cross-well validation

To further evaluate the generalization capability of the proposed CNN model, we employed the model to predict porosities from core CT images coming from a second well from the same field, and thus on data that was completely unseen by the model training process. This well penetrates only two of the previously mentioned formations, i.e., Formation 3 and Formation 4 as shown in Fig. 13. The lithology of Formation 4 in this well is very similar to the first well, consisting of mud and caliche rich marlstones. However, Formation 3 is quite different than the one in the first well, as it contains also intervals with more coarse-grained lithofacies. The obtained prediction results are presented in Fig. 13, where one can see quite good correlation between the predicted (light blue curve) and measured porosity (red circles), especially for depth intervals above XX17. As mentioned, the lithology in these intervals is very similar to the lithology in the first well. This might explain the higher prediction accuracy. However, the model predictions are less accurate for porosity values in the intervals where the lithology is significantly different than the lithologies that the model was trained on.

Overall, the MAE and r^2 between the core porosity measurements and model predicted porosity is 0.032 and 0.73, respectively. These results indicate the practical applicability of the proposed CNN model on unseen images from other wells.

7. Conclusions

This study investigated the capability of convolutional neural networks to automatically predict RCA porosity from two-dimensional whole core CT scan images and proposed a workflow to predict millimeter scale continuous porosity values. The proposed CNN regression model, trained to learn the relationship between convolution-derived features and RCA-derived porosity values, was compared with a simple linear regression model trained to predict RCA porosity from average gray-level attenuation computed for the same two-dimensional CT images.

Summarizing, the results show that the CNN model is well able to predict the porosity with a coefficient of determination of 0.81 for the unseen test set. This indicates that the optimized convolutional kernels have learned the textures and patterns that are important for estimating porosity. The capabilities of the CNN model in predicting porosity on completely unseen images, and the capabilities of cross-populating two wells with porosity values were moreover both tested quantitatively. The obtained results show that the model generalizes well on unseen images where the lithology is similar to those that the model has been trained on. However, images coming from significantly different lithologies, and images with specific features/image artefacts (e.g., images with drilling mud invasion and core barrel couplings) can result in model deficiencies.

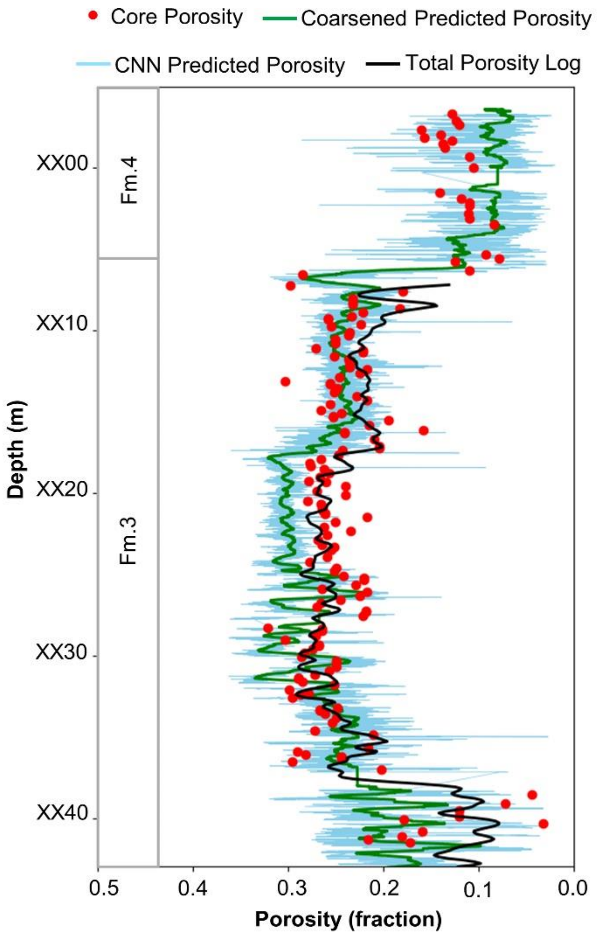

Fig. 3: Porosity prediction in the second well employing the proposed CNN model. The model performs quite well given that the lithology in some intervals is different than the lithofacies that the model is trained on.

Overall, comparison of the predicted porosity with total porosity log reveals that the proposed method can provide accurate millimeter scale porosity estimations, while the porosity log has a lower resolution and therefore overlooks such high-resolution variations.

Our conclusion is thus that the proposed method can be employed to estimate continuous core scale porosity values in an automatic fashion, and at early stages of the reservoir characterization process. This method can moreover be used to calibrate the porosity logs, thereby reducing the uncertainties associated with indirect calculations of the porosity from such logs. We also remark that it may be helpful to identify proper core plug locations for an improved core analysis. However, the most significant application of the proposed high-resolution porosity log might be in reservoir characterization and modeling.

It is worth to mention that in this study the CNN model was trained only on images from a single well. Ideally, training the model on multiple wells, covering different lithofacies and a wide range of porosity values, is expected to result in higher generalization capabilities.

Acknowledgements

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

Declarations

Funding

This study was supported by Equinor.

Conflict of interest

The authors declare that they have no conflict of interest.

Availability of data and material

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

Disclaimer

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

References

[1] Ketcham, R. A. and Carlson, W. D.: Acquisition, optimization and interpretation of X-ray computed tomographic imagery: Applications to the geosciences. *Computers and Geosciences*, *27*(4), 381–400 (2001). https://doi.org/10.1016/S0098-3004(00)00116-3

[2] Vinegar, H. J.: X-ray CT and NMR imaging of rocks. *Journal of Petroleum Technology*, *38*(03), 257–259 (1986). https://doi.org/10.2118/15277-PA

[3] Renter, J. A. M.: Applications of computerized tomography in sedimentology. *Marine Geotechnology*, *8*(3), 201–211 (1989) https://doi.org/10.1080/10641198909379868

[4] Ahmadi, M. A. and Chen, Z.: Comparison of machine learning methods for estimating permeability and porosity of oil reservoirs via petrophysical logs. *Petroleum*, *5*(3), 271–284 (2019). https://doi.org/10.1016/j.petlm.2018.06.002

[5] Ahmadi, M.-A., Ahmadi, M. R., Hosseini, S. M., and Ebadi, M.: Connectionist model predicts the porosity and permeability of petroleum reservoirs by means of petro-physical logs: Application of artificial intelligence. *Journal of Petroleum Science and Engineering*, *123*, 183–200 (2014). https://doi.org/10.1016/j.petrol.2014.08.026

[6] Aïfa, T., Baouche, R., and Baddari, K.: Neuro-fuzzy system to predict permeability and porosity from well log data: A case study of Hassi R'Mel gas field, Algeria. *Journal of Petroleum Science and Engineering*, *123*, 217–229 (2014). https://doi.org/10.1016/j.petrol.2014.09.019

[7] Al-Anazi, A. and Gates, I. D.: A Support Vector Machine Algorithm to Classify Lithofacies and Model Permeability in Heterogeneous Reservoirs. *Engineering Geology*, *114*(3–4), 267–277 (2010). https://doi.org/10.1016/j.enggeo.2010.05.005

[8] Al-Bulushi, N., King, P. R., Blunt, M. J., and Kraaijveld, M.: Development of artificial neural network models for predicting water saturation and fluid distribution. *Journal of Petroleum Science and Engineering*, *68*(3), 197–208 (2009). https://doi.org/10.1016/j.petrol.2009.06.017

[9] Al-Mudhafar, W. J.: Integrating well log interpretations for lithofacies classification and permeability modeling through advanced machine learning algorithms. *Journal of Petroleum Exploration and Production Technology*, *7*(4), 1023–1033 (2017). https://doi.org/10.1007/s13202-017-0360-0

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

[11] Helle, H. B., Bhatt, A., and Ursin, B.: Porosity and permeability prediction from wireline logs using artificial neural networks: A North Sea case study. *Geophysical Prospecting*, *49*(4), 431–444 (2001). https://doi.org/10.1046/j.1365-2478.2001.00271.x

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

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

[14] Rafik, B., and Kamel, B.: Prediction of permeability and porosity from well log data using the nonparametric regression with multivariate analysis and neural network, Hassi R'Mel Field, Algeria. *Egyptian Journal of Petroleum*, *26*(3), 763–778 (2017). https://doi.org/10.1016/j.ejpe.2016.10.013

[15] Rogers et al.: Predicting permeability from porosity using artificial neural networks. *AAPG Bulletin*, *79* (1995). https://doi.org/10.1306/7834DEFE-1721-11D7-8645000102C1865D

[16] Saljooghi, B. S. and Hezarkhani, A.: Comparison of WAVENET and ANN for predicting the porosity obtained from well log data. *Journal of Petroleum Science and Engineering*, *123*, 172–182 (2014). https://doi.org/10.1016/j.petrol.2014.08.025

[17] Wong et al.: Permeability determination using neural networks in the Ravva Field, Offshore India. *SPE Reservoir Evaluation and Engineering*, *1*(02), 99–104 (1998). https://doi.org/10.2118/38034-PA

[18] Wong, P. M., Gedeon, T. D., and Taggart, I. J.: An improved technique in porosity prediction: A neural network approach. *IEEE Transactions on Geoscience and Remote Sensing*, *33*(4), 971–980 (1995). https://doi.org/10.1109/36.406683

[19] Wood, D. A.: Predicting porosity, permeability and water saturation applying an optimized nearest-neighbour, machine-learning and data-mining network of well-log data. *Journal of Petroleum Science and Engineering*, *184*, 106587 (2020). https://doi.org/10.1016/j.petrol.2019.106587

[20] Zhang, G., Wang, Z., Li, H., Sun, Y., Zhang, Q., and Chen, W.: Permeability prediction of isolated channel sands using machine learning. *Journal of Applied Geophysics*, *159*, 605–615 (2018). https://doi.org/10.1016/j.jappgeo.2018.09.011

[21] Alqahtani, N., Armstrong, R. T., and Mostaghimi, P.: *Deep Learning Convolutional Neural Networks to Predict Porous Media Properties*. SPE Asia Pacific Oil and Gas Conference and Exhibition (2018). https://doi.org/10.2118/191906-MS

[22] Araya-Polo, M., Alpak, F. O., Hunter, S., Hofmann, R., and Saxena, N.: Deep learning–driven permeability estimation from 2D images. *Computational Geosciences*, *24*(2), 571–580 (2020). https://doi.org/10.1007/s10596-019-09886-9

[23] Bordignon, F., Figueiredo, L. P. D., Exterkoetter, R., Rodrigues, B. B., and Correia, M. D.: Deep Learning for Grain Size and Porosity Distributions Estimation on micro-CT Images. *Proceedings of the 16th International*

Congress of the Brazilian Geophysical Society & Expogef (2019).

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

[25] Kamrava, S., Tahmasebi, P., and Sahimi, M.: Linking morphology of porous media to their macroscopic permeability by deep learning. *Transport in Porous Media*, *131*(2), 427–448 (2020). https://doi.org/10.1007/s11242-019-01352-5

[26] Rabbani, A., Babaei, M., Shams, R., Da Wang, Y., and Chung, T.: DeePore: A deep learning workflow for rapid and comprehensive characterization of porous materials. *Advances in Water Resources*, *146*, 103787 (2020). https://doi.org/10.1016/j.advwatres.2020.103787

[27] Srisutthiyakorn*, N.: Deep-learning methods for predicting permeability from 2D/3D binary-segmented images. *SEG International Exposition and Annual Meeting* (2016). https://www.onepetro.org/conference-paper/SEG-2016-13972613

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

[29] Tembely, M. and AlSumaiti, A.: Deep Learning for a Fast and Accurate Prediction of Complex Carbonate Rock Permeability From 3D Micro-CT Images. *Abu Dhabi International Petroleum Exhibition and Conference* (2019). https://doi.org/10.2118/197457-MS

[30] Hunt, P. K., Engler, P., and Bajsarowicz, C.: Computed tomography as a core analysis tool: applications, instrument evaluation, and image improvement techniques. *Journal of Petroleum Technology*, *40*(09), 1203–1210 (1988). https://doi.org/10.2118/16952-PA

[31] Lopez, O., Berg, C. F., Rennan, L., Digranes, G., Forest, T., Kristoffersen, A., and Bøklepp, B. R.: Quick core assessment from CT imaging: From petrophysical properties to log evaluation. *In Int. Symp. Soc. Core Anal. Snowmass Colo. USA* (2016).

[32] LeCun, Y.: Generalization and network design strategies. Connectionism in perspective, 19, 143-155 (1989).

[33] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K.: Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[34] Anjos, C. E. M. dos, Avila, M. R. V., Vasconcelos, A. G. P., Neta, A. M. P., Medeiros, L. C., Evsukoff, A. G., and Surmas, R.: Deep learning for lithological classification of carbonate rock micro-CT images. *ArXiv:2007.15693* (2020) *[Cs, Eess]*. http://arxiv.org/abs/2007.15693

[35] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv:1207.0580* (2012) *[Cs]*. http://arxiv.org/abs/1207.0580

[36] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* (2014).

[37] Ioffe, S. and Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. *ArXiv:1502.03167 [Cs]* (2015). http://arxiv.org/abs/1502.03167

[38] Mustra, M., Delac, K., and Grgic, M.: Overview of the DICOM standard. *50th International Symposium ELMAR*, *1*, 39–44 (2008).

[39] Schneider, C. A., Rasband, W. S., and Eliceiri, K. W.: NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675 (2012). https://doi.org/10.1038/nmeth.2089

[40] Chollet, F. and others.: *Keras*. \url{https://Github.Com/Fchollet/Keras} (2015). https://keras.io/getting_started/faq/

[41] Chawshin, K., Berg, C. F., Varagnolo, D., Gonzalez, A., Heidari, Z., and Lopez, O.: Classifying lithofacies from textural features in whole core CT-scan images. SPE Res Eval & Eng. SPE-205354-PA (2021). https://doi.org/10.2118/205354-PA

[42] Baker, R. O., Yarranton, H. W., and Jensen, J. L.: 7—Conventional Core Analysis–Rock Properties. In R. O. Baker, H. W. Yarranton, and J. L. Jensen (Eds.),

Practical Reservoir Engineering and Characterization, 197–237 (2015). Gulf Professional Publishing. https://doi.org/10.1016/B978-0-12-801811-8.00007-9

[43] Ruder, S.: An overview of gradient descent optimization algorithms. *ArXiv:1609.04747 [Cs]* (2017). http://arxiv.org/abs/1609.04747

[44] Brownlee J.: Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Machine Learning Mastery (2016).

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

[46] Bergstra, J. and Bengio, Y.: Random Search for Hyper-parameter Optimization. *The Journal of Machine Learning Research*, *13*(null), 281–305 (2012).

[47] Hutter, F., Hoos, H. H., and Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, 507–523 (2011). https://doi.org/10.1007/978-3-642-25566-3_40

[48] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A.: Hyperband: a novel bandit-based approach to hyperparameter optimization. *ArXiv:1603.06560 [Cs, Stat]* (2018). http://arxiv.org/abs/1603.06560

[49] Snoek, J., Larochelle, H., and Adams, R. P.: Practical Bayesian optimization of machine learning algorithms. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, *2*, 2951–2959 (2012).

[50] Liu, F. T., Ting, K. M., and Zhou, Z.: Isolation forest. *Eighth IEEE International Conference on Data Mining*, 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17

[51] Liu, Fei Tony, Ting, K. M., and Zhou, Z.-H.: Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data*, *6*(1), 3:1-3:39 (2012). https://doi.org/10.1145/2133360.2133363