A Convolutional Neural Network approach to Electrical Resistivity Tomography

Mattia Aleardi°, Alessandro Vinciguerra°,*, Azadeh Hojat+,§

°University of Pisa, Earth Sciences Department, via S. Maria 53, 56126, Pisa, Italy

*University of Florence, Earth Sciences Department, via. G. La Pira 4, 50121, Florence, Italy

+Shahid Bahonar University of Kerman, Department of Mining Engineering, Kerman, 76188, Iran

§Politecnico di Milano, Department of Civil and Environmental Engineering, Piazza Leonardo da Vinci 32, 20133, Milano Italy

Corresponding author: Mattia Aleardi, mattia.aleardi@unipi.it

ABSTRACT

Electrical resistivity tomography (ERT) is an ill-posed and non-linear inverse problem commonly solved through deterministic gradient-based methods. These algorithms guarantee fast convergence toward the final solution but hinder accurate uncertainty assessments. On the contrary, numerical Markov Chain Monte Carlo algorithms provide accurate uncertainty appraisals but at the expense of a considerable computational effort. In this work, we develop a novel approach to ERT that guarantees an extremely fast inversion process and reliable uncertainty appraisals. The implemented method combines a Discrete Cosine Transform (DCT) reparameterization of data and model spaces with a Convolutional Neural Network. The CNN is employed to learn the inverse non-linear mapping between the DCT-compressed data and the DCT-compressed 2-D resistivity model. The DCT is an orthogonal transformation that here acts as an additional feature extraction technique that reduces the dimensionality of the input and output of the network. The DCT also acts as a regularization operator in the model space that significantly reduces the number of unknown parameters and the ill-

conditioning of the inversion procedure, thereby preserving the spatial continuity of the resistivity values in the recovered solution. The estimation of model uncertainties is a key step of geophysical inverse problems and hence we implement a Monte Carlo simulation framework that propagates onto the estimated model the uncertainties related to both noise contamination and network approximation (the so-called modeling error). We first apply the approach to synthetic data to investigate its robustness in case of erroneous assumptions on the noise and model statistics used to generate the training set. Then, we demonstrate the applicability of the method through inverting real data measured along a river embankment. We also demonstrate that transfer learning avoids retraining the network from scratch when the statistical properties of training and target sets are different. Our tests confirm the suitability of the proposed approach, opening the possibility to estimate the subsurface resistivity values and the associated uncertainties in near real-time.

KEYWORDS: Electrical Resistivity Tomography; Inversion; Convolutional Neural Networks.

INTRODUCTION

Electrical resistivity is an important property of geological formations with high sensitivity to fluid saturation and porosity and thus, the Electrical Resistivity Tomography (ERT) is a geophysical method widely and successfully employed for groundwater exploration, geotechnical characterization, mapping of contaminant plumes, landfill studies, and levees monitoring (see, for example, Legaz et al. 2009; Müller et al. 2010; Pollock and Cirpka 2012; Moradipour et al. 2016; Arosio et al., 2017; Crawford et al., 2018; Hojat et al. 2019a; Tresoldi et al. 2019; Hermans and Paepen, 2020; Aleardi et al. 2020a). The ERT inverse problem is nonlinear and ill-posed and is usually solved through deterministic gradient-based algorithms (Pidlisecky and Knight 2008; Karoulis et al. 2014) that linearize the problem around an initial solution thereby losing the information for accurate uncertainty appraisals. On the contrary, Markov Chain Monte Carlo algorithms can be employed to cast nonlinear inverse problems into a solid probabilistic framework

in which the final solution is the so-called posterior probability density (PPD) function in the model space (Sambridge and Mosegaard, 2002; Ramirez et al. 2005; Aleardi et al. 2018; Aleardi and Salusti, 2020; Pradhan and Mukerji, 2020; Aleardi et al. 2020b) that fully quantifies the ambiguities in the retrieved solution. However, the application of these methods to ERT inversion is usually hampered by both the high-dimensional parameter space and the expensive forward modeling operator. To partially mitigate these issues, model reparameterization techniques can be employed. These methods make use of different orthogonal basis functions (e.g., principal component analysis, wavelet transforms, Legendre polynomials, Discrete Cosine Transform) to reduce the dimensionality as well as the computational complexity of inverse problems. After such reparameterization, the unknown parameters become the numerical coefficients that multiply the basis functions (Dejtrakulwong et al., 2012; Lochbühler et al. 2014; Fernández Martínez et al. 2017; Aleardi 2019; Szabó and Dobróka, 2019). However, the compression should be applied keeping in mind that the model parameterization must always constitute a compromise between model resolution and model uncertainty (Grana et al. 2019).

In addition to the previously described conventional inversion algorithms, the recent advent of high-speed multi-core CPUs and GPUs over the last years has also promoted the applications of machine learning approaches (Monajemi et al., 2016; Goodfellow et al. 2016) to solve geophysical problems. In particular, Convolutional Neural Networks (CNNs) have recently gained attention (Krizhevsky et al. 2012) because they overcome some limitations of artificial neural networks such as local minima, overfitting, vanishing gradient, significant computational cost (Schmidhuber, 2015). Moreover, CNNs have been also implemented in many powerful software packages (Paszke et al. 2019). Training a CNN is a supervised learning task that requires a sufficiently large training set to iteratively refine and update the internal network parameters. This learning is an optimization process that minimizes a difference criterion between predicted and desired output. Even though the training is often computationally intensive, once the network is trained it converts an input dataset into the corresponding output response in real-time. In geophysics, CNNs have been initially applied to aid

structural interpretation of geophysical data such as seismic horizon and fault interpretation, and seismic texture identification (Xiong et al. 2018; Waldeland et al. 2018). Then, they have been extended to quantitatively solve many geophysical problems. For example, a CNN was employed by Lewis and Vigh (2017) and Richardson (2018) to full-waveform inversion, and by Araya-Polo et al. (2018) to seismic tomography. Wang et al. (2019) used a machine learning approach for seismic data interpolation. Park and Sacchi (2020) used a CNN for automatic velocity analysis; a trained convolutional network was employed by Das et al. (2019) and by Puzyrev (2019) for impedance inversion, and electromagnetic inversion, respectively. Moghadas (2020) used a CNN for inverting electromagnetic induction data, while Aleardi (2020a) exploited a CNN to speed up the Hamiltonian Monte Carlo sampling. A specific kind of fully convolutional neural network (the U-net) has been also employed to solve the ERT inversion (Liu et al. 2020). However, one well-known issue of machine-learning methods is that their performances sensibly worsen if the target and training data are significantly different (Goodfellow et al. 2016). To overcome this issue, transfer learning can be used (Li et al. 2020; Park and Sacchi, 2020), in which an additional training process with a small portion of target data is used to adjust the network internal parameters.

In this work, we train a CNN to map the apparent resistivity data to a 2-D subsurface resistivity model. The ERT is usually an under-determined problem with more unknowns than data points. For this reason, we use the Discrete Cosine Transform to compress the data and model spaces and to reduce the number of unknown parameters. The use of the DCT reparameterization in the model space also guarantees that realistic spatial variabilities are preserved in the retrieved solution. Indeed, the order of the retained non-zero DCT coefficients determines the wavelength of the recovered resistivity model. The resistivity models forming the training and validation sets are drawn from a previously defined a-priori resistivity distribution that incorporates a 2-D stationary Gaussian variogram expressing the assumed lateral and vertical variability of the resistivity values in the study area. A Finite-Elements (FE) code (Karoulis et al. 2013) is used to compute the observed data associated with each generated model. The CNN inversion is combined with a Monte Carlo (MC) simulation to

estimate the uncertainties affecting the retrieved solution. To this end, we propagate onto the model space not only the uncertainties related to noise contamination but also the so-called modeling error introduced by the CNN. Indeed, the trained network learns an approximated function that maps the observed data into the associated model and this approximation introduces an additional source of uncertainty.

We first focus on synthetic inversion experiments in which we assess the applicability and the robustness of the CNN inversion in the presence of errors in the assumed noise statistic and a-priori resistivity distribution. For simplicity, we assume log-Gaussian distributed resistivity values in the synthetic experiments. Therefore, the simple kriging geostatistical method (Azevedo and Soares, 2017) is used to generate the resistivity models forming the training and validation examples. Then, the CNN predictions and the MC estimated uncertainties are compared with the outcomes of a more computationally expensive MCMC inversion running in the DCT space (Vinciguerra et al. 2020) and with the predictions yielded by a local least-squares inversion algorithm (Loke, 2018). Finally, we apply the implemented approach to field data and its outcomes are compared with those provided by the local inversion. In this case, transfer learning is applied to update the internal parameters of the network previously trained for the synthetic data application. We also take into account the facies-dependent behavior of the resistivity values and hence, we assume a non-parametric mixture prior model in which each mode is associated with a given litho-fluid class. As far as the authors are aware, this is the first time that DCT compression, CNN inversion, and MC simulations are combined to solve the ERT inversion and to estimate model uncertainties.

METHODS

Discrete Cosine Transform

The DCT is a linear and orthogonal transformation that projects an N-length signal (e.g., vector of model parameters) to an N-length vector containing the coefficients of N different cosine (base) functions. This approach concentrates most of the information of the original signal into the low-

order DCT-coefficients so that only $q<N$ coefficients can be used to accurately approximate the input signal. In the context of geophysical inversion, this means that the numerical values of these q DCT coefficients become the unknowns to be inferred from the data. Estimating the retained DCT-coefficients reduces the parameter space dimensionality and can significantly improve the computational efficiency of the inversion procedure. We use the DCT parameterization because it exhibits superior compression power over other compression methods (Lochbühler et al. 2014).

Several variants of DCT exist with slightly modified definitions, but in this work, we use the so-called DCT-2 formulation that is the most common one. Hereafter we simply refer to the DCT-2 transformation as the DCT. This is a Fourier-related transform that uses only real numbers to express a finite signal in terms of the sum of cosine functions oscillating at different frequencies. If we assume a 2-D resistivity model $\boldsymbol{\rho}(x,y)$ in which x=[0,1,…,M_x-1] and y=[0,1,…,M_y-1] represent the horizontal and vertical coordinates, respectively, the associated 2-D transform is defined as follows:

$$\begin{cases} \mathbf{R}_{k_x,k_y} = \frac{1}{\sqrt{M_x}} \cdot \frac{1}{\sqrt{M_y}} \sum_{x=0}^{M_x-1} \sum_{y=0}^{M_y-1} \boldsymbol{\rho}(x,y), \qquad if k_x = k_y = 0 \\ \mathbf{R}_{k_x,k_y} = \sqrt{\frac{2}{M_x}} \sqrt{\frac{2}{M_y}} \sum_{x=0}^{M_x-1} \sum_{y=0}^{M_y-1} \boldsymbol{\rho}(x,y) cos\left(\frac{(2x+1)\pi k_x}{2M_x}\right) cos\left(\frac{(2y+1)\pi k_y}{2M_y}\right), if k_x, k_y \neq 0 \end{cases}, \quad (1)$$

where $\mathbf{R}_{k_x,k_y}$ represents the k_x-th and k_y-th DCT coefficient. The values within the matrix $\mathbf{R}$ represent the unknowns to be estimated in a DCT-reparameterized inverse problem. Equation 1 can be compactly rearranged in matrix form:

$$\mathbf{R} = \mathbf{B}_y \boldsymbol{\rho} \mathbf{B}_x^T, \quad (2)$$

where $\mathbf{B}_x$ and $\mathbf{B}_y$ are the matrices with dimensions $M_x \times M_x$ and $M_y \times M_y$, respectively that contain the DCT basis functions, whereas the $M_y \times M_x$ matrix $\mathbf{R}$ expresses the DCT coefficients. Most of the spatial variability of the resistivity model is explained by low-order DCT coefficients and for this reason, an approximation of the subsurface resistivity model can be obtained as follows:

$$\tilde{\boldsymbol{\rho}} = \left(\mathbf{B}_y^q\right)^T \mathbf{R}_{qp} \mathbf{B}_x^p, \qquad (3)$$

where $\tilde{\boldsymbol{\rho}}$ is the approximated $[M_y \times M_x]$ resistivity model, $\mathbf{B}_y^q$ is a $[q \times M_y]$ matrix containing only the first q rows of $\mathbf{B}_y$; $\mathbf{B}_x^p$ is a $[p \times M_x]$ matrix containing only the first p rows of $\mathbf{B}_x$, whereas the matrix $\mathbf{R}_{qp}$ represents the first q rows and p columns of $\mathbf{R}$. In other words, the scalar q and p represent the retained number of basis functions along the y and x directions used to derive the approximated resistivity model. Therefore, the DCT transformation reduces the $(M_y \times M_x)$-D full resistivity model space to a $(q \times p)$-D DCT-compressed domain with $p < M_x$ and $q < M_y$.

In the context of CNN, the DCT constitutes an additional feature extraction technique that reduces the number of pixels in the input and output images of the CNN, thus reducing the complexity of the CNN architecture (i.e., number of hidden layers) needed to map the input of the network into the corresponding output. This translates into an easier CNN configuration and hyperparameter settings and a faster training phase because fewer CNN parameters must be updated. In the context of ERT inversion, the DCT acts as a regularization operator in the model space that mitigates the ill-conditioning of the inverse problem, while preserving reasonable spatial resistivity variations in the estimated model. Figure 1 shows some DCT basis functions of different orders in a 2-D space. Note that the variability of the solution along each dimension is directly determined by the orders of the retained DCT coefficients. Finally, we refer the interested readers to Lochbühler et al. (2014) for a comparison of different parameterization techniques in the context of ERT inversion.

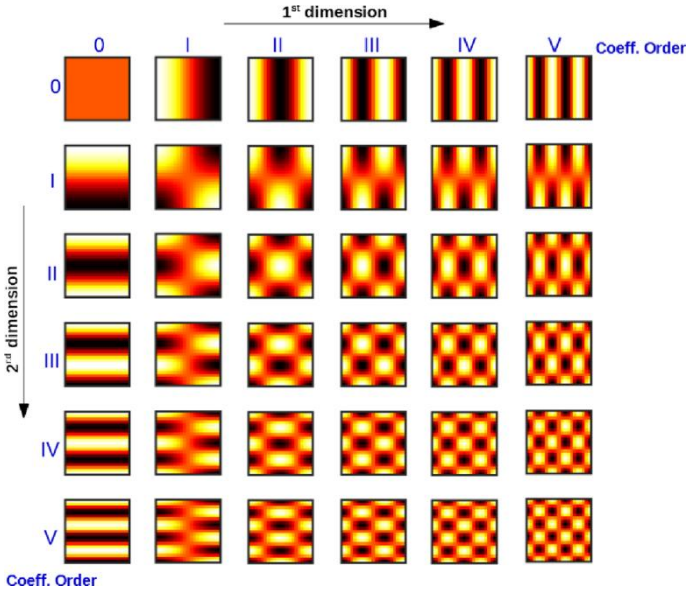


Figure 1: 2-D DCT basis functions of different orders. Dark and light colors code low and high numerical values, respectively.

Convolutional neural networks

The relation between a CNN and its generated model is usually expressed as follows:

$$\mathbf{O} = F(\mathbf{P}, \mathbf{L}), \quad (4)$$

where F denotes the CNN as a function that maps the input $\mathbf{L}$ to the output $\mathbf{O}$ through the CNN internal parameters $\mathbf{P}$. CNNs use blocks of convolutional layers, subsampling layers, and fully connected layers, to extract features from 1-D, 2-D, or 3-D input maps treated as grids of pixels. The extracted features form the so-called feature maps. The core of CNNs is the convolutional layer, in which the feature maps are convolved with convolution filters. This process can be written as:

$$O_j^p = f\left(b_j + \sum_{i=1}^{I} O_i^{p-1} * W_j\right), \quad j = 1, 2, \ldots, J \quad (5)$$

where I represents the number of the feature maps in the (p-1)-th layer, whereas J is the total number of feature maps in the p-th layer, which is equal to the number of filters considered in that layer; b_j is a scalar value representing the j-th bias of the p-th layer, $*$ represents the convolution operator, $f()$ is the so-called activation function used to include non-linearity in the mapping process, O_j^p is the j-th feature map in the p-th layer, O_i^{p-1} represents the i-th feature map in the (p-1)-th layer, and W_j denotes

187 the *j*-th convolutional filter of the *p*-th layer. This filter has a user-specified size and slides over the
188 input map with a specified stride. The internal CNN parameters to be updated are the values
189 associated with the filters W_j and the biases b_j in each layer.

190 The aim of the subsampling layers (also known as pooling) is to prevent overfitting by reducing the
191 dimension of the feature map generated in the convolutional layer and the number of features. The
192 most common pooling strategies are max-pooling and average pooling (Scherer et al. 2010). After
193 the features of the input image are extracted by convolutional layers, they are usually fed into fully
194 connected layers, which are appended to the end of the last convolutional blocks.

195 At the first iteration, the internal CNN parameters are initialized and then updated during the
196 iterative learning process. The learning process aims to minimize an error (loss) function that
197 measures the difference between the desired and the computed output. A back-propagation algorithm
198 usually drives the minimization process and the updating of the filter values. This updating process
199 can be written as follows:

200 $$\mathbf{P}_i = \mathbf{P}_{i-1} - \gamma \frac{\partial \varepsilon}{\partial \mathbf{P}_{i-1}}, \quad (6)$$

201 where *i* represents the iteration number, ε is the loss function value, and γ is the so-called learning
202 rate, and **P** again represents all the CNN learnable parameters.

203 To define the CNN architecture, some hyperparameters must be set: number of hidden layers and
204 number of filters, kernel width and stride of the convolution and pooling operators, activation
205 function, a method for weight initialization, optimization algorithm to minimize the loss function and
206 to update the filter weights, number of epochs. There are no rigid rules to set these hyperparameters
207 and the final choice is often dictated by personal preference and experience. We found the optimal
208 setting through a trial and error procedure in which, different hyperparameters are changed (i.e.,
209 number of filters, filter size, learning rate, batch size, and the type of activation function) and the final
210 net architecture has been determined based on the net performances on the validation set. In the case

that different CNNs achieve similar prediction capabilities, we selected the one with the fastest learning process.

APPLICATION TO SYNTHETIC DATA

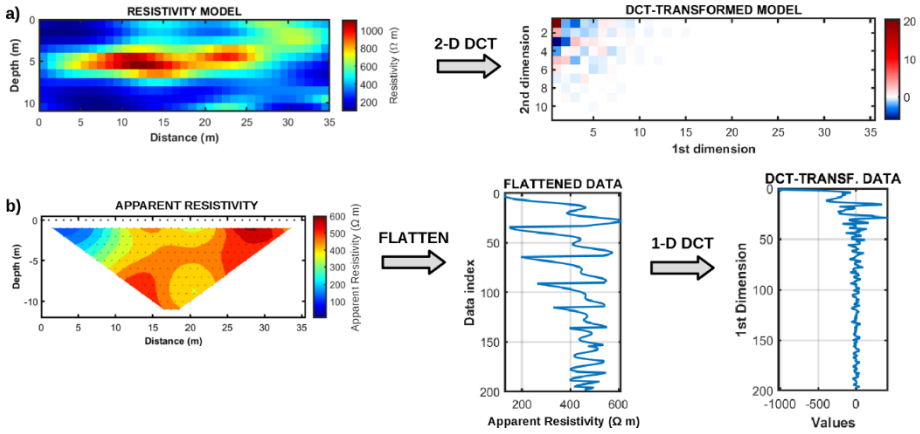
The implemented CNN inversion

We assume that the study area is 35 m long and 11 m deep. This area is discretized with quadratic cells with spatial dimensions of 1 m × 1 m. We also assume that the subsurface resistivity values in the target area follow a stationary (i.e., spatial invariant) a-priori log-Gaussian distribution, with a spatial variability pattern defined by a 2-D Gaussian variogram model. The ranges of the assumed variogram model are 3 m and 8 m along the vertical and horizontal directions, respectively. We employ a log-normal distribution to reduce the variability range of the resistivity values. The statistical properties of this prior model are summarized in Table 1 and they can be inferred from available borehole data or geologic information in practical applications. To generate the training and validation sets, we use the simple kriging geostatistical method to randomly draw 20000 resistivity models from the assumed prior statistic. The previously mentioned FE code has been used to compute the data for each generated model. Since most field data that are used in our researches are measured with the Wenner array, we simulate a Wenner acquisition layout using 36 electrodes and an injected current of 1 Ampere. The maximum a value we considered is 11. This configuration results in $11 \times 35 = 385$ model parameters to be estimated from 198 data points. To the so obtained 20000 data vectors we add Gaussian uncorrelated noise with a standard deviation equal to 10 % of the average standard deviation of the noise-free pseudosections associated with the 20000 generated models (see Table 1). The training set includes 18000 out of the 20000 examples, while the remaining 2000 examples form the validation set.

Observed data characteristics		**Prior resistivity model**	
Uncorrelated noise standard deviation	10 % of *n*	Type of prior distribution	Log-Gaussian
Correlated noise standard deviation	0 % of *n*	Prior mean resistivity value [ln(Ω·m)]	5.82
Type of noise distribution	Gaussian	Prior standard deviation of resistivity [ln(Ω·m)]	0.86
		Vertical range of the Gaussian variogram model	3 m
		Lateral range of the Gaussian variogram model	8 m

Table 1: Principal characteristics of the observed data and of the prior model that have been used to generate the training and validation sets. *n* indicates the average standard deviation of the noise-free datasets computed from the generated models.

The next step after the generation of the training and validation examples involves the estimation of the optimal number of DCT coefficients needed to approximate the resistivity models and the associated datasets. The optimal number of DCT coefficients in the data and model spaces are determined by decomposing some of the 20000 models and data previously generated. Figure 2 shows an example of a DCT-projected resistivity model and the associated data drawn from the training examples. Note that because of its trapezoidal shape, the apparent resistivity pseudosection cannot be expressed as a 2-D matrix, and thus, it has been flattened to a 1D vector before the DCT projection.



248

249 Figure 2: a) Example of a resistivity section extracted from the prior distribution and the
250 associated DCT projection. b) Example of an apparent resistivity pseudosection and the
251 associated DCT projection. Note that the apparent resistivity pseudosection is flattened to a
252 1D vector before the DCT projection.

253 Figure 3 illustrates, for four models extracted from the prior distribution, the explained variability
254 of the original uncompressed model as the number of the considered basis functions along the two
255 DCT dimensions increases. The explained variability is computed as the ratio between the standard
256 deviation of the approximated and the uncompressed resistivity model (Aleardi, 2020b). We observe
257 that 5 coefficients along the 1st DCT dimension and 4 along the 2nd dimension explain almost 100 %
258 of the total variability of the original, un-compressed resistivity models. This means that the spatial
259 resolution of the recovered model is equal to that expressed by the assumed variogram model and that
260 the DCT compression does not sensibly reduce the resolution of the result because the number of
261 retained coefficients allows recovering almost the total variability of prior realizations.

262 Figure 4 represents the explained variability as the number of DCT coefficients increases for the
263 four data associated with the models previously shown in Figure 3. In this case, we observe that 150
264 basis functions explain almost the total variability of the original four datasets. Based on the previous
265 considerations, we approximate the resistivity model using the first 4 rows and 5 columns of the
266 associated DCT matrices, while only the first 150 DCT coefficients of the flattened apparent
267 resistivity pseudosection are considered in the data domain. The use of DCT reduces the 385-D full

model space to a 20-D space, while the 198-D data domain has been compressed to a 150-D space. In the implemented approach, the DCT transformation acts as an additional feature extraction technique that reduces both the number of unknown parameters to invert for and the dimensionality of the input and the output of the network. Therefore, the use of the DCT reduces the complexity of the CNN architecture (i.e. number of hidden layers) needed to convert the input into the associated output response. This translates into an easier CNN hyperparameter setting and a faster training phase because fewer CNN parameters must be adjusted.

To better understand the effect of the DCT compression of the model space, Figure 5 compares an uncompressed resistivity model drawn from the prior distribution with the approximated models derived when different numbers of DCT coefficients are considered. If only two coefficients are considered along the two DCT dimensions, the approximation provides just a very smoothed version of the original model, while only 5 coefficients along the 2 DCT dimensions guarantee a satisfactory reproduction of the lateral and vertical variations of the model.

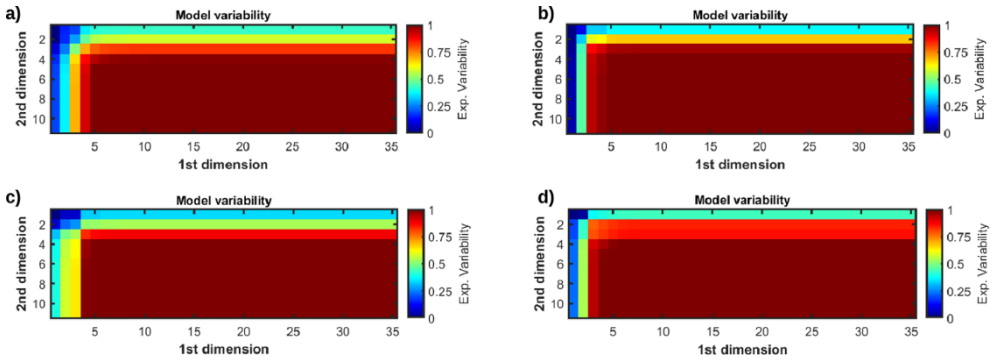
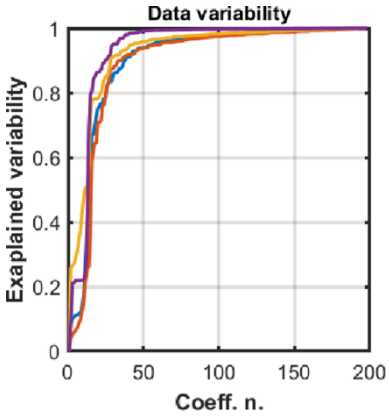


Figure 3: Examples of the explained model variability as the number of DCT coefficients along the 1st and 2nd DCT dimension increases. a), b), c), and d) illustrate the explained variability for four different models extracted from the prior distribution. In each plot, the numerical value with coordinates (x, y) indicates the explained variability if the first x, and y DCT coefficients along the 1st and 2nd DCT dimensions, respectively, are used for compressing the resistivity model. In all cases, it emerges that 5 DCT coefficients along the

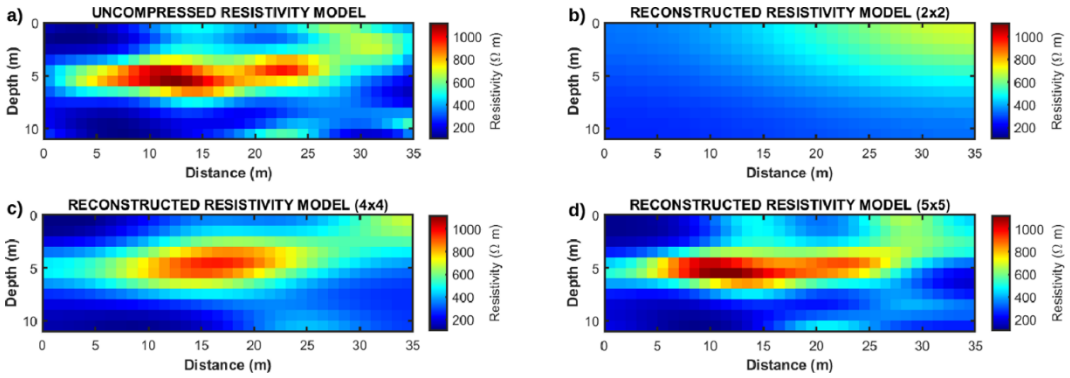
289 1st dimension and 4 along the 2nd dimension explain almost 100 % of the variability of the
290 uncompressed resistivity models.



291

292 Figure 4: Explained variability as the number of DCT coefficients increases for the data
293 associated with the four resistivity models considered in Figure 3. Different colors refer to
294 different data.

295



296

297 Figure 5: a) Original uncompressed resistivity model drawn from the prior distribution. b)
298 Approximated model when only the first 2 DCT coefficients along the first and the second
299 DCT dimensions are considered. c) Approximated model when the first 4 DCT coefficients
300 along the first and the second DCT dimensions are considered. d) Approximated model when
301 the first 5 DCT coefficients along the first and the second DCT dimensions are considered.

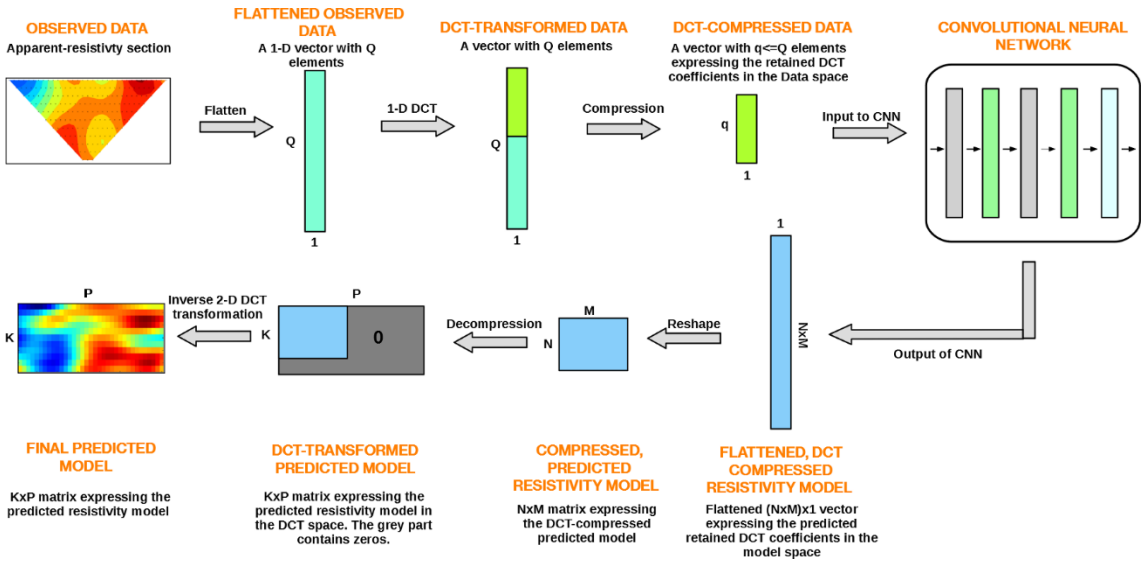


Figure 6: Schematic representation of the CNN-ERT inversion framework. The Monte Carlo error propagation is not included in this figure. See the text for details.

Figure 6 describes the workflow of the implemented CNN inversion that retrieves the resistivity model from the pseudosection. One additional but crucial step is the assessment of the uncertainty affecting the estimated solution. We must project onto the model space both the noise affecting the resistivity data and the so-called modeling error introduced by the CNN approximation. In the following discussion, we describe our Monte Carlo approach to quantify and project such errors: Let **M** represent the ensemble of resistivity models forming the training dataset, while **N** is the associated ensemble of models predicted by the trained CNN. A sample of the modeling error can be obtained as $\mathbf{E} = \mathbf{M} - \mathbf{N}$ (Hansen and Cordua, 2017)**.** Assuming a Gaussian distribution, the modeling error can be defined as $\mathcal{N}(0, \mathbf{C}_e)$, where $\mathbf{C}_e$ is the covariance of **E.** This error together with the noise term $\mathcal{N}(0, \mathbf{C}_n)$ (also assumed Gaussian-distributed) are propagated onto the final prediction with an iterative MC approach. Now, let **d** be the vector expressing the observed data input to the CNN, whereas *n* represents the number of MC simulations. The implemented MC approach for uncertainty propagation comprises the following six steps:

1) Use the trained CNN to compute the predicted resistivity model $\mathbf{m_b}$ from the observed data vector **d**;
2) Run a forward modeling to compute the noise-free $\mathbf{d_b}$ data associated to $\mathbf{m_b}$;
3) For *i*=1 to *Q*
    a. Draw $\mathbf{n_i}$ from $\mathcal{N}(0, \mathbf{C}_n)$ and compute $\mathbf{d_{n,i}} = \mathbf{d_b} + \mathbf{n_i}$;
    b. Use the trained CNN to compute the predicted model $\mathbf{m_{n,i}}$ from $\mathbf{d_{n,i}}$;
    c. Draw $\mathbf{e_i}$ from $\mathcal{N}(0, \mathbf{C}_e)$ and compute $\mathbf{m_{e,i}} = \mathbf{m_{n,i}} + \mathbf{e_i}$;
    d. Store $\mathbf{m_{e,i}}$

Each generated $\mathbf{m_{e,i}}$ model can be considered a possible subsurface model in agreement with the observed data, the trained CNN, and the assumed distributions for the noise and modeling errors. This ensemble of *Q* models can be used to numerically derive the statistical properties of the posterior model such as posterior mean and posterior standard deviation. For simplicity, we assume that both error terms (noise and modeling errors) are Gaussian, but the implemented approach can be applied to whatever parametric or non-parametric error distribution. Note that the previous MC approach is extremely fast because the network predicts a model from the input data (steps 1 and 4) in real-time.

Setting the CNN architecture

The CNNs usually consist of one input layer, one or more hidden convolutional layers, one or several fully connected layers, and one output layer. In our case, the training ensemble is constituted by a tensor of $150 \times 1 \times 18000$ where 18000 is the number of the training examples and 150 is the number of the retained DCT basis functions in the data space. The corresponding output is a vector of $20 \times 1 \times 18000$ DCT coefficients where 20 is the number of DCT basis functions used to compress the resistivity model. We perform different experiments to optimally set the main CNN hyperparameters, and the final CNN architecture (Figure 7) was chosen according to the best fit on the validation set. It consists of two convolution blocks and a fully connected layer. The first

convolution block uses 5 1-D convolution filters of size 3, and a stride of 1, whereas the second block includes 10 1-D convolution filters of size 5, and a stride of 1. After each convolutional layer, we use the LeakyRelu activation function with a slope of 0.1 (Krizhevsky et al., 2012). Batch normalization is used within each convolution block because Santurkar et al. (2018) suggested that it would guarantee more predictive and stable behavior of the gradients as well as faster training. After the convolutional blocks, max-pooling of size 2 and a stride 1 is applied for subsampling. Before the fully connected layer, a dropout of 0.1 is used to prevent overfitting. We adopt the RMSprop optimizer (i.e., an unpublished, adaptive learning rate method) running for 20 epochs to minimize the root-mean-square error (RMSE) between the expected and the predicted outputs. The He method is used to initialize the network weights (He et al., 2015). We use a batch size of 32, and an initial learning rate of 0.001 that is multiplied by 0.9 every epoch. We chose this batch size and initial learning rate because they guaranteed the best performances in the experiments we carried out. In particular, the benefits of small batch sizes have been discussed in many studies (e.g., Masters and Luschi 2018). From the evolution of the RMSE error for the training and the validation datasets, we observe that the learning process successfully converges in less than 5 epochs (Figure 8).

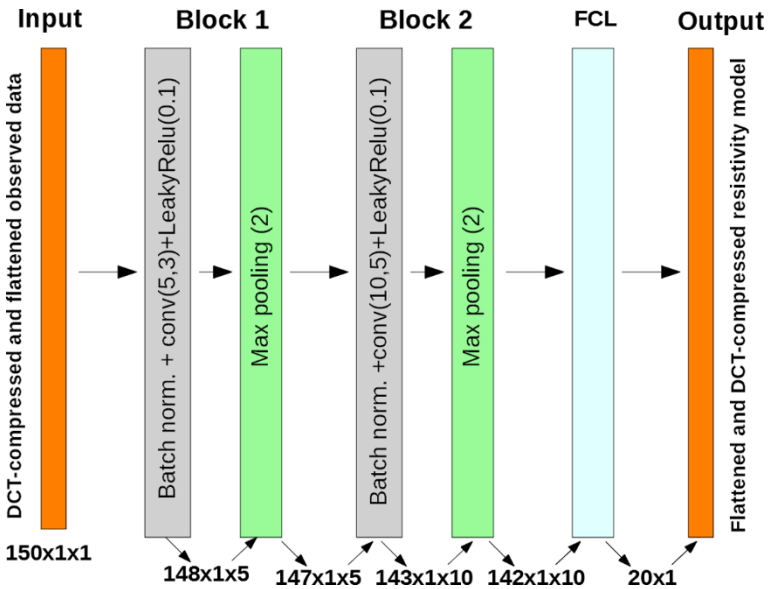


Figure 7: Schematic representation of the adopted 1-D CNN architecture. The first image on the left is the input of the network that is a vector expressing the 150 DCT coefficients used

for data compression. The image on the far right is the output of the network that is a vector containing the 20 DCT coefficients coding the resistivity model. The intermediate rectangles represent convolutional layers and are annotated with key parameters. In the grey rectangles, the initial value in brackets (e.g., 5) indicates the number of filters. This is followed by the filter size (i.e., 3). Within the green rectangles, we also indicate the dimension of the max-pooling filter (2). The cyan rectangle represents the fully connected layer (FCL). The numbers on the bottom of each rectangle indicate the dimension of the input and output to each layer.

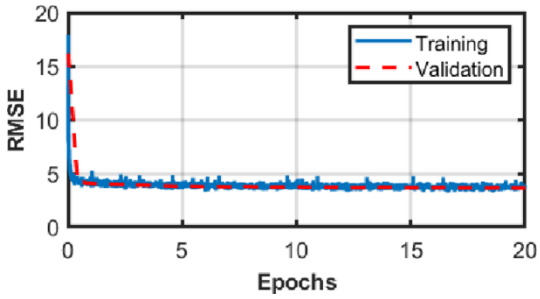


Figure 8: Evolution of the root-mean-square error in the training and validation sets during training.

We now discuss the CNN performances when some of the hyperparameters are changed. In the following examples, only one parameter at a time is modified to the previously described network architecture. As the test set, we employed a synthetic model of a high-resistivity half-space that hosts a low-resistivity rectangular block. This model will be described more in detail in the next section. Figure 9a shows the results obtained by changing the number of convolutional layers. As expected, it is not guaranteed that adding more layers will improve the performance. Indeed, redundancy results in increased training time, waste of computational power, and numerical instability. In this case, two convolution blocks constitute the best compromise between the network performances and the computational effort. Figure 9b shows the RMSE on the training, validation, and test datasets for a different number of filters (i.e., feature maps) in the first convolutional block. It emerges that 5 filters guarantee the best performances using an RMSE loss function and that adding more layers does not

improve the final error value. This probably means that the essential features extracted by the DCT are already predicted by a simple network and that a more complex CNN architecture only results in a waste of computational resources. To decrease the final RMSE value, more DCT coefficients should be considered in the model space but at the expense of an increased dimensionality of the network output. However, we will demonstrate at the end of this section that the considered number of DCT coefficients guarantees accurate reconstruction of the subsurface models. Figure 9c illustrates the network performances for different dimensions of the training set. We can observe that 20000 training samples offer the best compromise between the generalization capability of the network (expressed by the RMSE value on the validation and test datasets) and the computational cost needed for training, although similar performances can be also achieved with only 10000 training examples. In this regard, if we consider serial Matlab codes running on a common notebook equipped with a quad-core intel core i-7 7700HQ CPU@2.80 GHz with 16 Gb RAM, the selected CNN architecture can be trained in approximately 2 minutes, whereas all the tests in Figure 9 run in approximately 22 minutes. These small computing times are guaranteed by the limited dimensions of the input and the output of the network and by the simple CNN architecture employed. In other words, the use of the DCT compression allows a fast assessment of the performances of CNNs with different hyperparameter settings and architectures, thus limiting the human effort and the computational resources needed to configure the network. Note that the vast majority of the network configurations considered in Figure 9 provide similar RMSE values. This means that different networks achieve similar predictions and that the quality of the results is not heavily affected by the selected network configuration.

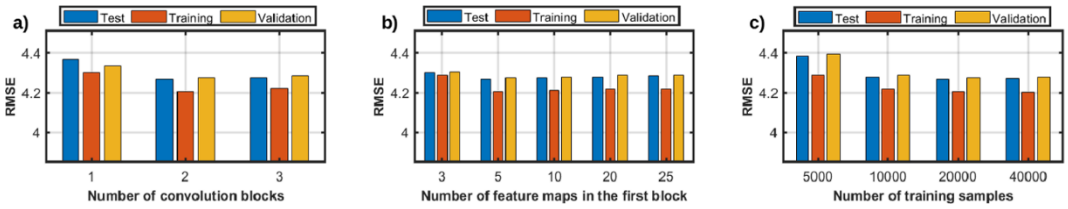


Figure 9: RMSE errors on the training, validation, and test sets when some CNN hyperparameters are changed. a) Changing the number of convolution blocks. b) Changing the number of feature maps (filters) in the first block. c) Changing the dimension of the training set.

Finally, Figure 10 shows some comparisons between resistivity models extracted from the validation set and the corresponding CNN predictions. As expected, we observe that moving from the shallowest to the deepest part of the subsurface and from the center to the lateral edges of the model, the quality of the predictions worsens. These examples illustrate that the resistivity values below 6 m depth are not informed by the data. Therefore, the estimation of these unknowns from the recorded data is a hopelessly ill-conditioned problem regardless of any approach we may use. This comparison shows that the trained CNN can reliably predict the resistivity values within the first 5-6 m depth. The previous results refer to resistivity models and associated data that perfectly honor the statistical characteristics assumed in the learning phase. Therefore, in the following section, we assess the robustness of the CNN-ERT inversion to errors in the assumed noise statistic and prior model. The CNN predictions will also be validated against those provided by a deterministic inversion and by an MCMC algorithm sampling the DCT-compressed model space (Vinciguerra et al. 2020).

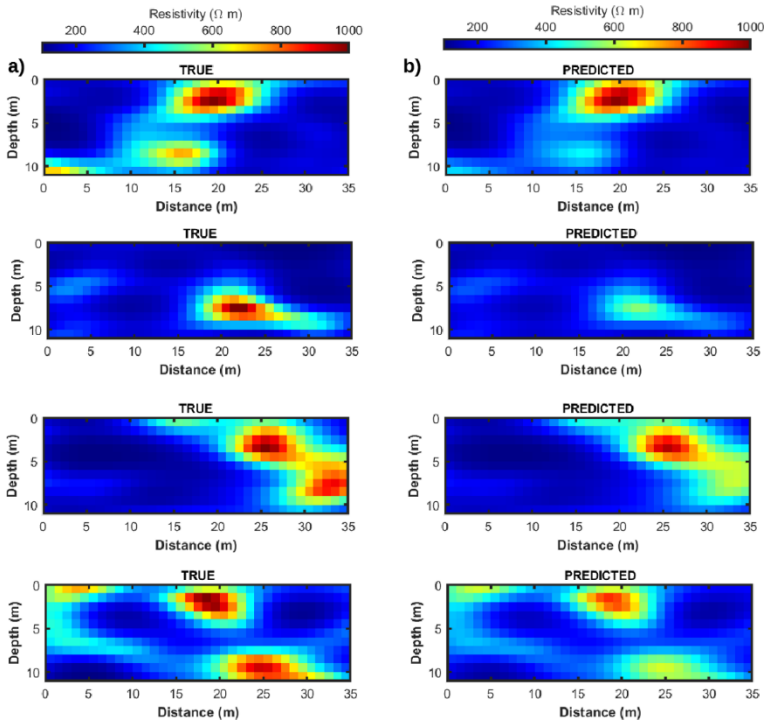


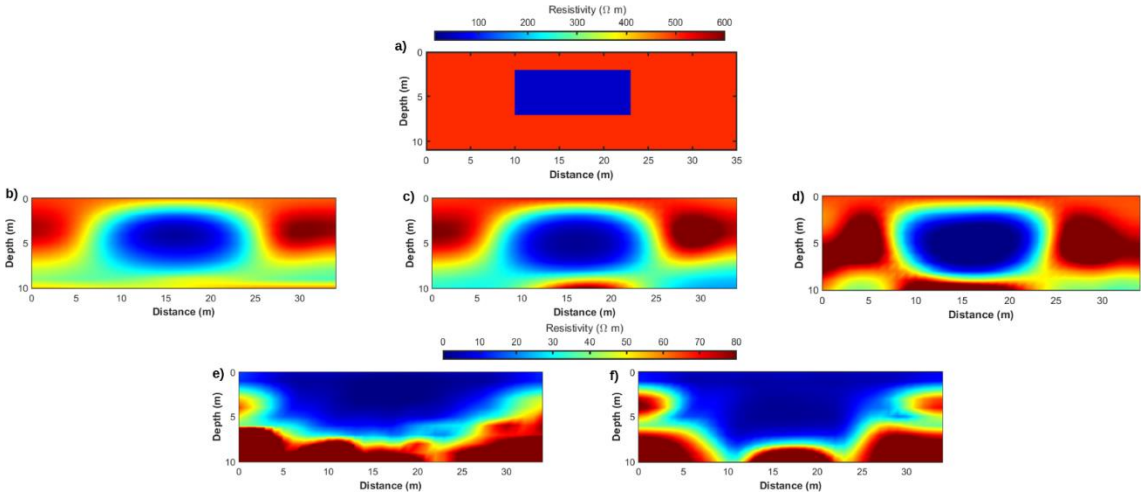
Figure 10: a) Some models extracted from the validation set and (b) the corresponding CNN predictions.

Inversion

In all the following tests the reference model used to derive the observed data has the same dimension as the models considered in the training phase and is constituted by a low-resistivity rectangular block hosted in a high-resistivity homogeneous half-space (Figure 11a). Note that for this model, the spatial variability of the resistivity values and their statistical distribution are very different from those assumed in the learning process (i.e., a log-Gaussian prior model with a Gaussian variogram). In the following, we also assess the accuracy and the stability of the CNN predictions when the distribution of the noise affecting the data and the statistical resistivity properties (i.e., mean and variance) in the true, target model differ from those assumed during the training phase.

However, before analyzing the robustness of the CNN-ERT inversion, we validate its predictions (in terms of both estimated model and uncertainties) with those achieved by a computationally demanding MCMC algorithm sampling the DCT-compressed model space. In this context, note that

the unknown parameters for both the CNN and MCMC inversions are the retained 20 DCT coefficients in the model space. In this first example, the mean and the variance of the true resistivity values are equal to those of the training set, although their statistical distribution and spatial variability are different (Figure 11a). The previously mentioned FE code was used to compute the observed data that are contaminated with Gaussian uncorrelated noise with statistical properties equal to those previously considered in the learning phase (i.e., a standard deviation equal to 10% of the standard deviation of the noise-free dataset). The CNN prediction is also compared with that yielded by a deterministic least-squares approach. In Figures 11b-11d we note that the CNN inversion retrieves a final solution very close to the a-posteriori mean estimated by the MCMC sampling and with the model provided by the deterministic inversion. In all cases, the background resistivity values and the low resistivity anomaly are well recovered and the anomaly is also properly located. As expected, the quality of the predictions decreases at the lateral edges and bottom of the model due to the limited parameter illumination. The slightly lower spatial resolution of the CNN prediction with respect to deterministic inversion results can be ascribed to the fact that abrupt lateral and vertical resistivity contrasts are not modeled by the employed variogram model assumed for the network training. Figures 11e and 11f compare the estimated standard deviation values affecting the retrieved solution estimated by the implemented MC algorithm and by the MCMC inversion, respectively. In both cases, the low resistivity anomaly is recovered with high accuracy, while the cells located at the lateral edges and below 6 m depth are associated with high uncertainties. These results demonstrate the reliability of the proposed inversion framework. Indeed, the CNN+MC approach provides final predictions and uncertainty estimations in agreement with those yielded by the much more computationally demanding MCMC inversion.



462

463 Figure 11: a) The true model for this first example. Note that the mean and the standard
464 deviation of the resistivity values are equal to those of the training set, but the statistical
465 distribution and the spatial variability of the resistivity values differ from those assumed in
466 the learning phase. b) CNN prediction. c) Mean posterior solution provided by the MCMC
467 sampling. d) Resistivity model estimated by the deterministic inversion. e) Standard
468 deviation values computed from 10000 MC simulations. f) MCMC estimate of the posterior
469 standard deviation.

470 Figure 12 shows some examples of MC simulated resistivity models that were used to compute the
471 standard deviation represented in Figure 10e. Again, as expected, the differences in the simulated
472 models increase at the lateral edges and bottom of the study area. As a final consideration, we point
473 out that the CNN result of Figure 11b and the MC estimation of the standard deviation shown in
474 Figure 11e can be computed in near real-time (i.e., the model and the uncertainties are almost
475 instantaneously estimated from the data; see the discussion section for more details), while the
476 MCMC algorithm takes almost 10 hours and thousands of forward modeling runs to attain stable
477 posterior model assessments. The deterministic inversion (Figure 11d) takes less than 2 minutes to
478 converge but the local linearization hinders accurate uncertainty assessments.

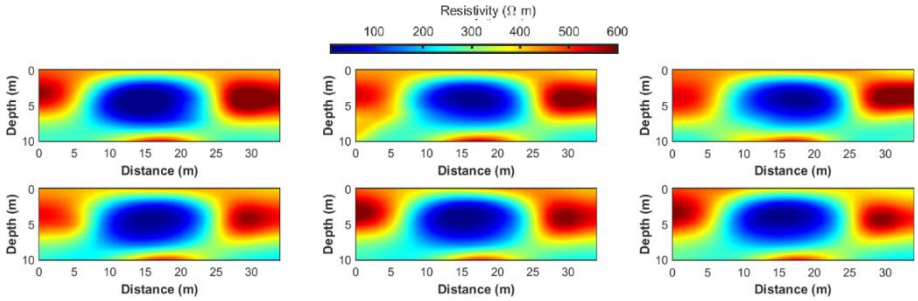


Figure 12: Some examples of MC simulations used to compute the standard deviation shown in Figure 11e.

The effects of errors in the assumed noise statistic

Now we analyze the robustness of the CNN predictions in case of erroneous assumptions on the statistical properties of the noise contaminating the observed data. We remind that in the training phase we considered Gaussian uncorrelated noise with a standard deviation value equal to 10% of the standard deviation value of the noise-free dataset. We perform eight inversion tests in which the observed data are contaminated with both uncorrelated and correlated Gaussian noise with different statistical properties. The true model in these experiments is the same previously used for the comparison with the MCMC inversion (Figure 11a). Therefore, in these tests the mean and the standard deviation of the resistivity values in the target model are equal to those of the training examples. In the first four tests (Tests 1-4) we only consider uncorrelated noise contaminating the measured pseudosections but with increasing standard deviation values (i.e., decreasing Signal to Noise, S/N, ratios in the observed data). Figure 13 compares the noise-free pseudosection associated with the true model, and the pseudosections contaminated with 20% and 50% of Gaussian uncorrelated noise (Figures 13b and 13c, respectively) that constitute the observed datasets for Tests 1 and 4, respectively. In Tests 5-8 we add both spatially correlated and uncorrelated Gaussian noise to the data. The details of these eight inversion experiments are given in Tables 2 and 3.

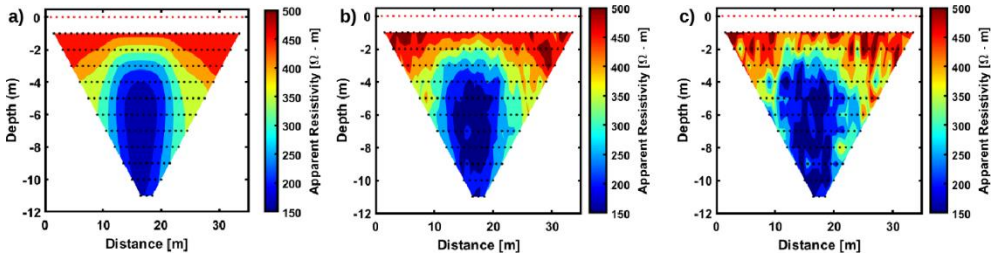


Figure 13: a) Noise-free pseudosection associated with the reference synthetic model. b) Pseudosection contaminated with 20% of Gaussian uncorrelated noise. c) Pseudosection contaminated with 50% of Gaussian uncorrelated noise. b), and c) represent the observed datasets for Tests 1 and 4, respectively.

Parameters	**Test 1**	**Test 2**	**Test 3**	**Test 4**
Uncorrelated noise standard deviation	20 % of *n*	30 % of *n*	40 % of *n*	50 % of *n*
Correlated noise standard deviation	0 % of *n*	0% of *n*	0 % of *n*	0 % of *n*
Error on the prior mean	0% of *m*	0% of *m*	0% of *m*	0% of *m*
Error on the prior standard deviation	0% of *s*	0% of *s*	0% of *s*	0% of *s*

Table 2: Error on the assumed noise and model statistics for Tests 1-4. *n* indicates the standard deviation of the noise-free observed dataset; *m* and *s* are respectively the mean and the standard deviation values of the prior resistivity model assumed in the learning phase.

Parameters	Test 5	Test 6	Test 7	Test 8
Uncorrelated noise standard deviation	20 % of *n*	30 % of *n*	40 % of *n*	50 % of *n*
Correlated noise standard deviation	10 % of *n*	20 % of *n*	30 % of *n*	50 % of *n*
Error on the prior mean	0% of *m*	0% of *m*	0% of *m*	0% of *m*
Error on the prior standard deviation	0% of *s*	0% of *s*	0% of *s*	0% of *s*

Table 3: Error on the assumed noise and model statistics for Tests 5-8. *n* indicates the standard deviation of the noise-free observed dataset; *m* and s are respectively the mean and the standard deviation of the a-priori model assumed in the learning phase. The range values of the correlated noise are 3 m and 8 m along the vertical and horizontal directions, respectively.

In Figure 14, as expected, we observe that the quality of the predictions worsens as the noise increases, or in other terms, as the overestimation of the S/N ratio increases. In all cases, the low resistivity anomaly is correctly located but the inversion tends to underpredict the actual resistivity contrast between the rectangular block and the homogeneous half-space as the difference between the actual and the assumed noise standard deviation increases. In Tests 1-3, the predicted data computed on the estimated model show good matches with the observed data, although their differences increase as the overestimation of the actual S/N ratio increases. The data match significantly decreases in Test 4 compared to Tests 1-3.

Figure 15 illustrates the results for Tests 5-8. In the case of low correlated and uncorrelated noise contamination, the CNN still provides quite accurate model estimations in which the low resistivity anomaly is correctly located (Tests 5-6). As previously observed in Figure 14, the predicted resistivity contrasts underestimate the actual contrasts as the overestimation of the S/N ratio increases (Tests 5-

7). As the correlated noise increases, the quality of the results decreases and some biased predictions appear; for example, the low resistivity anomaly is not recovered from the data (Test 8). In this last case, the predicted data does not match the observed one. However, the previous inversion tests demonstrated that the CNN inversion is quite robust against errors in the assumed noise statistic, and only a significant overestimation of the actual S/N ratio produces unrealistic and biased predictions.

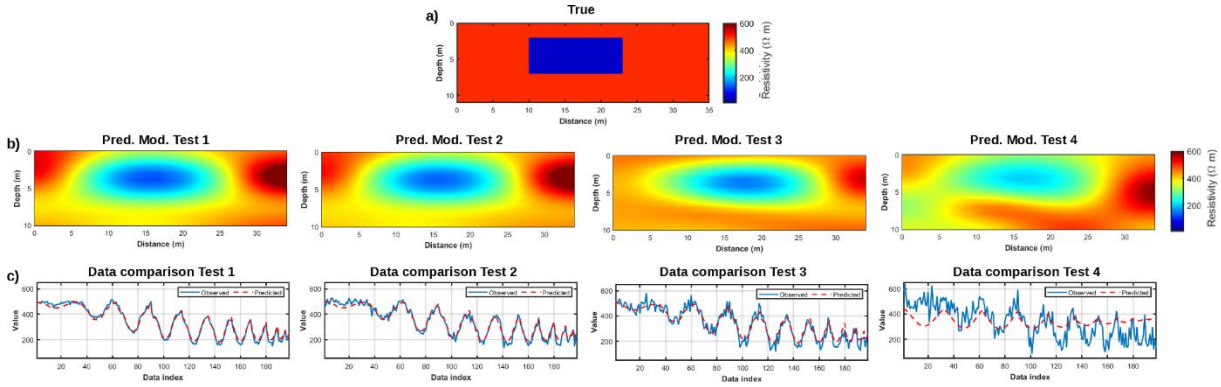


Figure 14: a) The true model for Tests 1-4. b) CNN predictions for the different tests. c) Comparison between the observed data (blue curves) and the data derived from the CNN prediction for different tests (red curves). The horizontal axis in c) represents the index associated with each apparent resistivity value along the data vector.

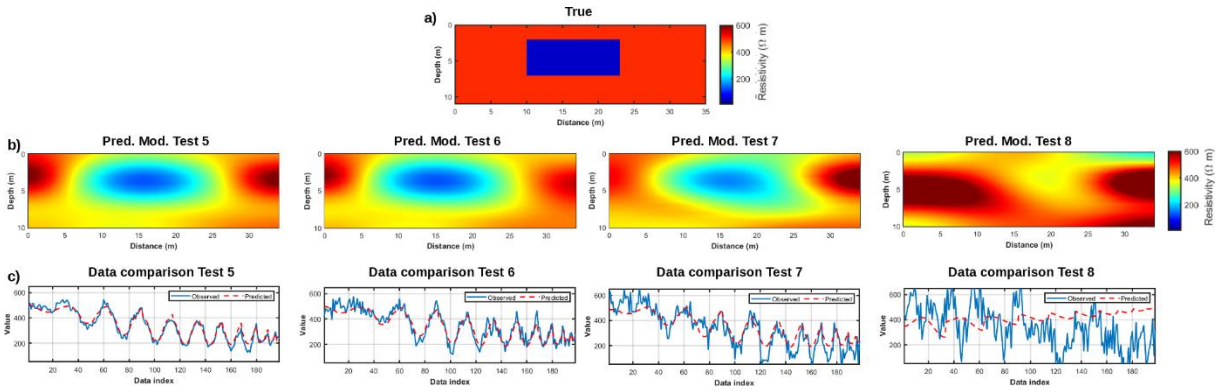


Figure 15: a) The true model for Tests 5-8. b) CNN predictions for the different tests. c) Comparison between the observed data (blue curves) and the data derived from the CNN prediction for different tests (red curves).

The effects of erroneous assumptions on the noise and model statistics

We analyze the combined effect of erroneous assumptions on both the noise and the model statistics on the CNN predictions. We again perform eight inversion tests employing the previously trained network. Similar to Tests 1-4, the observed data in the first four experiments (Tests 9-12) are contaminated with uncorrelated Gaussian noise characterized by different standard deviation values, but in this case, the mean and the variance values of the target, true model differ from those assumed in the learning phase. We simulate that the assumed prior mean and standard deviation (of the log-transformed resistivity values) underestimate the actual values in the study area. In particular, the errors in the assumed mean and standard deviation values in Tests 9-12 linearly increased from 5 % to 20 % (see Table 4 for details).

Parameters	**Test 9**	**Test 10**	**Test 11**	**Test 12**
Uncorrelated noise standard deviation	20 % of *n*	30 % of *n*	40 % of *n*	50 % of *n*
Correlated noise standard deviation	0 % of *n*	0% of *n*	0 % of *n*	0 % of *n*
Error on the prior mean	5 % of *m*	10 % of *m*	15 % of *m*	20 % of *m*
Error on the prior standard deviation	5 % of *s*	10 % of *s*	15 % of *s*	20 % of *s*

Table 4: Error on the assumed noise and model statistics for Tests 9-12. *n* indicates the standard deviation of the noise-free observed dataset; *m* and s are respectively the mean and the standard deviation of the prior log-Gaussian model assumed in the learning phase.

Figure 16 shows the results for Tests 9-12. For errors on the model statistic lower than 15 % the CNN inversion is still able to provide good results in which the low resistivity anomaly is correctly located and the actual resistivity values are satisfactorily recovered. In these cases (Tests 9-10), the observed data are well reproduced by the predicted model. The quality of the CNN predictions significantly

decreases when the errors on the prior model assumptions are equal to 15 %, and the true model is not recovered for errors on the prior assumptions equal to 20 %. In this case (Test 12), the low resistivity body is not recovered and significant low and high-resistivity artifacts appear in the CNN solution; as a consequence, the recovered model does not reproduce the observed data. These tests demonstrate that the CNN inversion is much more sensitive to errors in the prior model statistic than to errors in the assumed noise properties.

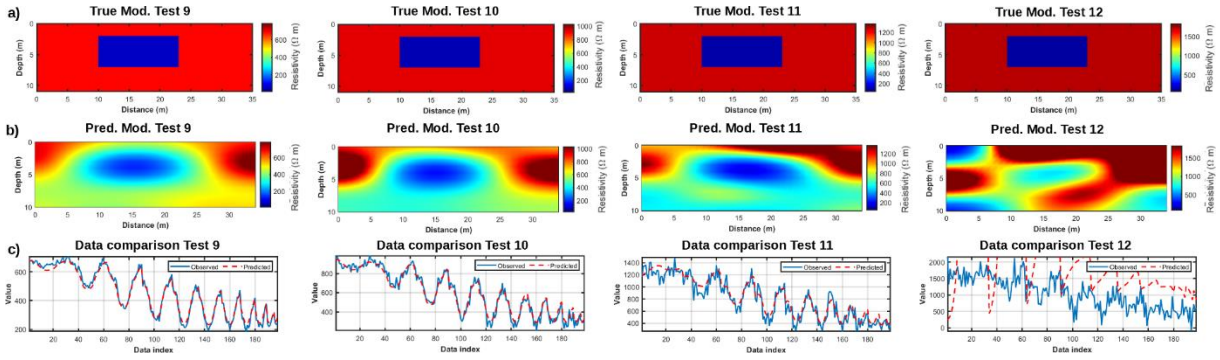


Figure 16. a) The true models for Tests 9-12. Note the different color scales. b) CNN predictions for different tests. c) Comparison between the observed data (blue curves) and the data derived from the CNN prediction for different tests (red curves).

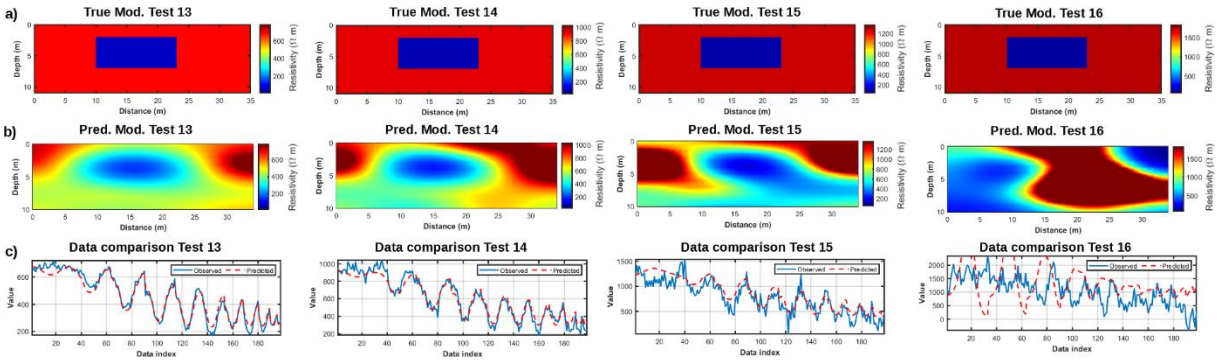
Finally, in Tests 13-16 we combine erroneous model assumptions with both uncorrelated and correlated Gaussian noise affecting the data (see Table 5 and Figure 17). For model errors lower than 10 %, the CNN still achieves satisfactory predictions in which the low resistivity rectangle is correctly located and the actual resistivity values are retrieved. Differently, when severe coherent noise contamination in the data adds to prior model errors higher than 15 % (Tests 15 and 16), the actual resistivity distribution and the geometries of the subsurface model are not retrieved by the CNN inversion. In these cases, the observed data are poorly predicted.

Figure 18 summarizes the results of Tests 1-16 and represents the L2 norm difference between the true model and the CNN solution. As previously mentioned, the CNN inversion seems to be quite robust in case of erroneous assumptions on the noise statistic unless significant underestimations of the coherent noise occur (Tests 4, 7, and 8). Differently, the quality of the CNN predictions is much

582 more affected by errors in the assumed prior model. The CNN results are poor in case of significant
583 overestimations of the S/N ratio and for errors higher than 15 % in the prior model assumptions (e.g.,
584 Tests 11, 12, 15, and 16).

Parameters	**Test 13**	**Test 14**	**Test 15**	**Test 16**
Uncorrelated noise standard deviation	20 % of *n*	30 % of *n*	40 % of *n*	50 % of *n*
Correlated noise standard deviation	10 % of *n*	20 % of *n*	30 % of *n*	50 % of *n*
Error on the prior mean	5 % of *m*	10 % of *m*	15 % of *m*	20 % of *m*
Error on the prior standard deviation	5 % of *s*	10 % of *s*	15 % of *s*	20 % of *s*

585 Table 5: Error on the assumed noise and model statistics for Tests 13-16. *n* indicates the
586 standard deviation of the noise-free observed dataset; *m* and s are respectively the mean and
587 the standard deviation of the prior model used in the learning phase. The range values of the
588 correlated noise are 3 and 8 m along the vertical and horizontal directions, respectively.



589

590 Figure 17: a) The true models for Tests 13-16. Note the different color scales. b) CNN
591 predictions for different tests. c) Comparison between the observed data (blue curves) and
592 the data derived from the CNN prediction for different tests (red curves).

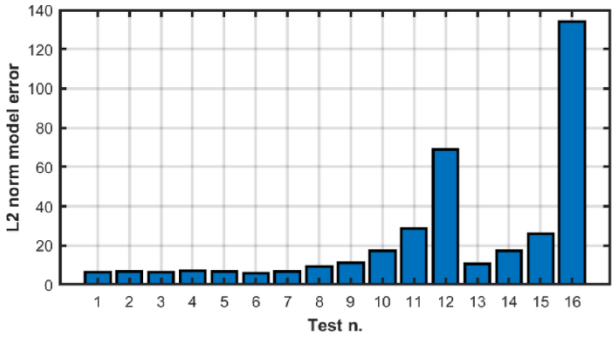


Figure 18: L2 norm differences between the true model and the CNN predictions for the 16 inversion tests.

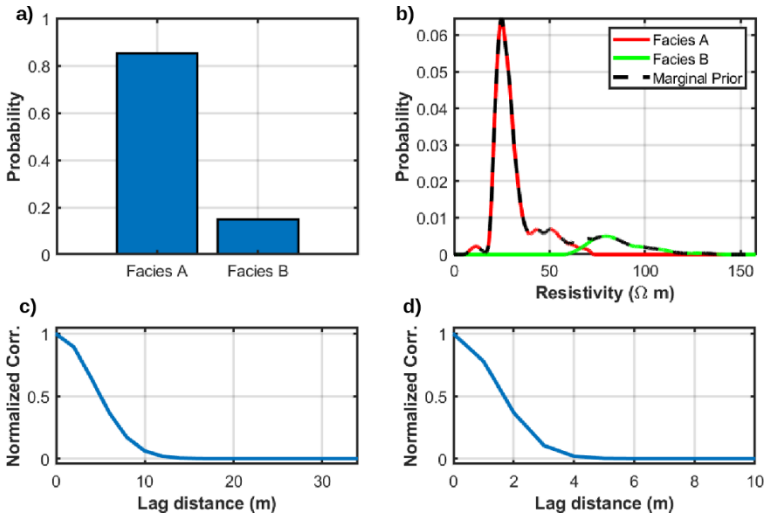
APPLICATION TO FIELD DATA

We apply the implemented CNN inversion to field data acquired by a permanent monitoring system installed along a river embankment. In this work, we limit to invert a single dataset and we refer the reader to Hojat et al. (2019b) for more information about the study site. The electrode layout is buried in a 0.5 m-deep trench and thus we used the data corrected for the effect of the soil overlaying the electrodes (Hojat et al. 2019c; Hojat et al. 2020). The inversion covers an area that is 94 m long and 14 m deep and was discretized with rectangular cells with dimensions of 1 m and 2 m along the vertical and horizontal directions, respectively. A few comparison measurements were performed in the reconnaissance phase of this project to compare the Wenner, Wenner-Schlumberger, and dipole-dipole arrays. Having obtained similar results, the Wenner array was selected to ensure a good signal-to-noise ratio (Dahlin and Zhou 2004). Therefore, the acquisition layout comprises a Wenner acquisition geometry with 48 electrodes with unit electrode spacing of 2 m. This configuration results in 705 resistivity values to be estimated from 360 data points. To define the a-priori distribution we exploit both the available geological information about the investigated area and the multiple data and the associated inverted resistivity sections obtained during the permanent monitoring. In particular, we expect a clay body hosting a more permeable layer constituted by sand and gravel located at around 2-3 m depth. Therefore, we assume two different facies (i.e., facies A and B) associated with the clay and sand formations and characterized by low and high resistivity values,

614 respectively. The prior facies probabilities are equal to $p(\boldsymbol{\pi} = \text{facies } A) = 0.85$ and $p(\boldsymbol{\pi} =$
615 facies $B) = 0.15$, where $\boldsymbol{\pi}$ indicates the facies variable. Differently from the previous examples, we
616 now assume a non-parametric prior model that properly takes into account the expected facies
617 dependency of the resistivity values. This prior can be written as follows:

618 $$p(\boldsymbol{\rho}) = \sum_{k=1}^{K} \omega_k \, p(\boldsymbol{\rho}|\boldsymbol{\pi}_k), \quad (7)$$

619 where K=2, $\omega_1 = 0.85$, and $\omega_2 = 0.15$. In this application, the extensive geological information
620 available helped us in the definition of an appropriate prior model. The prior can be computed, for
621 example, by applying the kernel density estimation algorithm (Parzen, 1962) to the available data
622 about the study area (core samples, and resistivity models inverted at different times). We also assume
623 a 2-D stationary Gaussian variogram model with vertical and lateral ranges equal to 2 m and 6 m,
624 respectively. Figures 19a and 19b represent the prior assumption on the resistivity values, whereas
625 Figures 19c and 19d illustrate the assumed lateral and vertical correlation functions coding the
626 Gaussian variogram. Note that due to the limited variations of the resistivity values, no logarithmic
627 transformation is applied here.
628



629

630 Figure 19: a) Expected frequency of occurrence of the two facies in the investigated area.
631 Facies A and B refer to clay and sand/gravel, respectively. b) Non-parametric marginal prior

distribution for the resistivity values (dotted black curve). The red and green curves show the non-parametric components associated with each facies. c), and d) normalized spatial correlation functions coding the expected variability along the lateral and vertical directions, respectively.

We take the previously trained network, but given the different prior assumptions, transfer learning is used to update the network weights. Indeed, transfer learning avoids retraining a network from scratch when it is applied to different datasets. This approach takes a pre-trained network and uses it as a starting point to learn a new task. Fine-tuning a network with transfer learning is usually much faster and easier than training a network with randomly initialized weights, thereby allowing a quick transfer of the learned features to a new task using a smaller number of training examples. When using transfer learning, it is important to decide which part of the already trained CNN model must be updated and this usually depends on the difference between the target and the training data (Park and Sacchi 2020). After some tests (not shown here for brevity) we decided to update only the parameters associated with the second convolutional block and the fully-connected layer. To this end, we generate from the prior 500 training examples (Figure 20), while 50 models form the validation ensemble. The Direct Sequential Simulation (DSS) method is used to generate the training and validation datasets, whereas the same FE code previously employed in the synthetic experiments is used to compute the associated datasets.

To set the optimal number of DCT coefficients to retain we use the same strategy previously described in the synthetic experiments. In this case, it turns out that 15 and 10 DCT coefficients along the first and second DCT dimensions explain almost the total variability of the generated models (Figures 21a-d), whereas 200 DCT basis functions are enough to accurately approximate the pseudo sections (Figure 21e). Therefore, the DCT reduces the 705-D uncompressed model space to a 150-D compressed space, while the 360-D data space is reduced to a 200-D space.

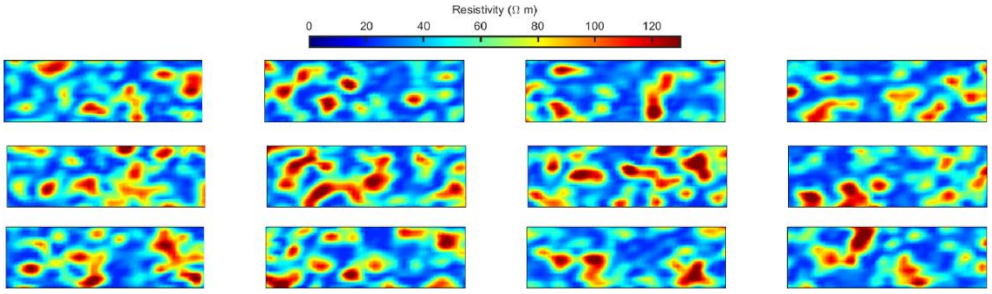


Figure 20: Some examples of resistivity models drawn from the prior distribution.

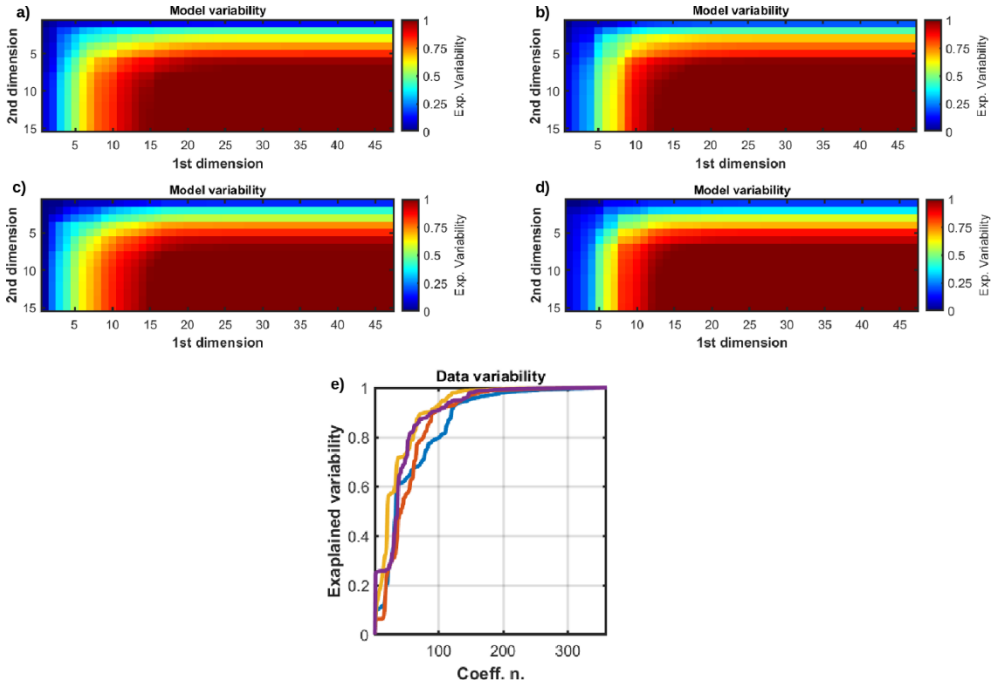
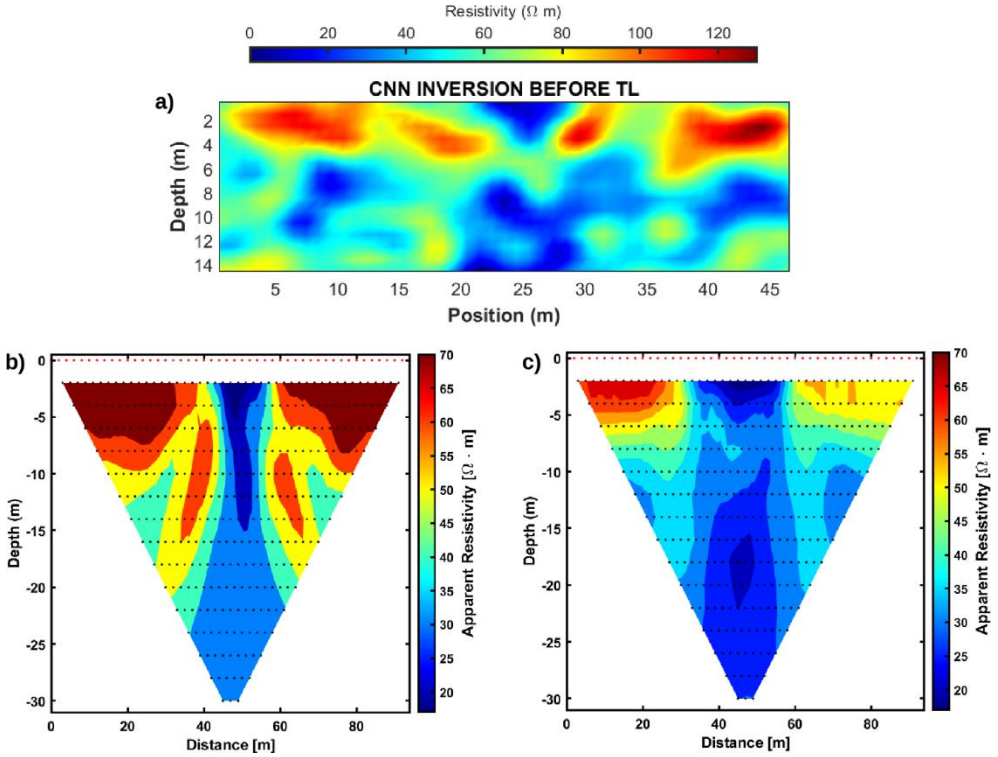


Figure 21: a)-d) Examples of explained model variability as the number of DCT coefficients along the 1st and 2nd DCT dimension increases. a), b), c), and d) illustrated the explained variability for four different models extracted from the prior distribution. In each plot, the numerical value with coordinates (x, y) indicates the explained variability if the first x, and y DCT coefficients along the 1st and 2nd DCT dimensions, respectively, are used for compressing the resistivity model. In all cases, it emerges that 15 DCT coefficients along the 1st dimension and 10 along the 2nd dimension explain almost 100 % of the variability of the uncompressed resistivity models. e) Explained variability as the number of DCT coefficients increases for the data associated with the four resistivity models considered in

669 a)-d). Different colors refer to different data. It results that 200 coefficients explain almost
670 100 % of the variability.

671



672

673 Figure 22: a) Resistivity model predicted by the trained CNN before the application of
674 transfer learning (TL). b) Predicted pseudosection computed on the model shown in a). c)
675 Observed pseudosection.

676

677

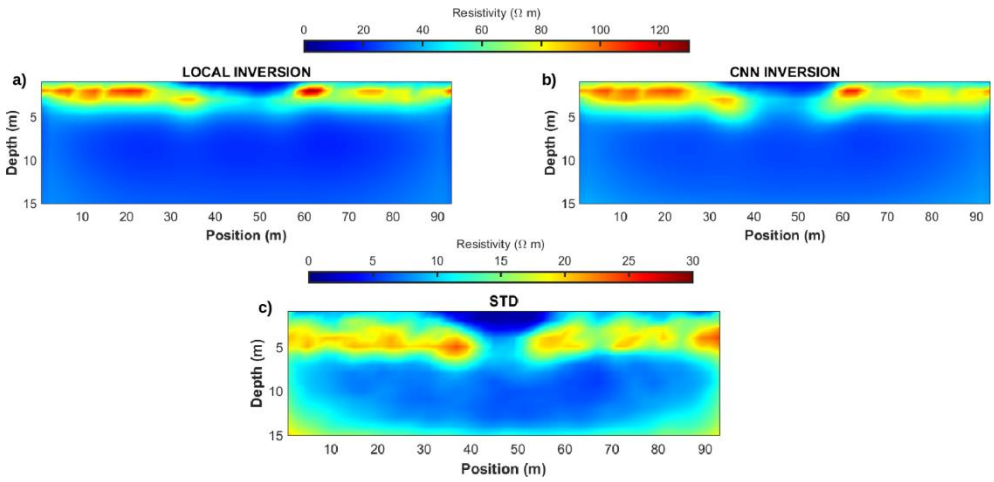


Figure 23: a) The model estimated by the gradient-based least-squares inversion. b) The predicted model provided by the CNN inversion. c) The standard deviation map estimated through 10000 MC simulations.

As an example, Figure 22a shows the CNN prediction before transfer learning. The poor match between the observed and predicted pseudosections (Figures 22b and 22c) is an indication of the low accuracy of the result. In particular, the overprediction of the observed apparent resistivity values proves that the model of Figure 22a tends to overpredict the resistivity over the study area. Figure 23a shows the result provided by a local least-squares inversion approach (Loke, 2018), whereas Figure 23b illustrates the model estimated by the implemented CNN algorithm after transfer learning. The similarity of the two inversion outcomes proves the reliability of the final solution and also confirms the suitability of the proposed algorithm for real data applications. The slightly lower spatial resolution of the CNN prediction is again related to the DCT compression of the model space. As expected from previous inversion results obtained in the same area and from the available geological information, both algorithms predict a high resistivity body around 2 m depth (associated with sand/gravel) hosted in a low resistivity medium (clay). Different from the local approach, the implemented inversion also provides an estimate of the uncertainty affecting the retrieved solution that is represented in Figure 23c in the form of a standard deviation section. As expected, the

shallowest part of the subsurface is characterized by the lowest uncertainty while the ambiguity increases within the high resistivity formation, at the lateral edges and deepest part of the model due to the lower illumination. Some examples of the 10000 MC realizations used to numerically compute the standard deviation section are shown in Figure 24. Note that all the realizations univocally predict a high resistivity body located around 2 m depth. Figure 25 compares the observed data and the two pseudosections computed from the model estimated by the local inversion and from the CNN solution. Both inversions provide similar predicted data characterized by a good match with the measured pseudosection. The significant improvement of the match between the observed data and the pseudosection computed on the model estimated after transfer learning highlights both the importance of the prior assumptions for proper training and that transfer learning can be conveniently applied when the statistical properties of training and target sets differ.

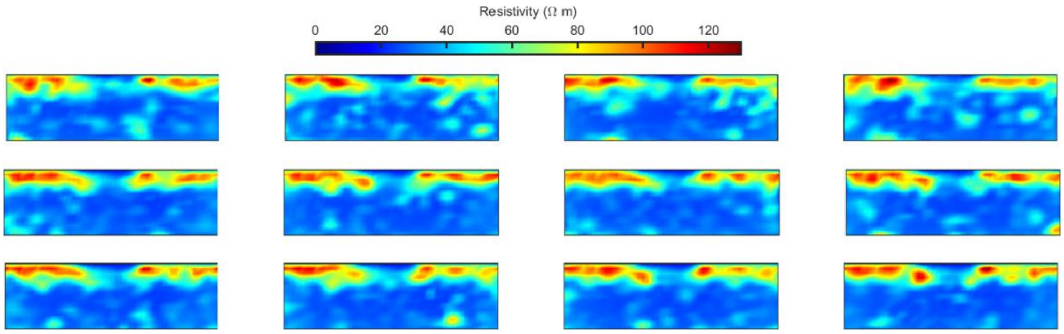


Figure 24: Some examples of MC simulations used to compute the standard deviation shown in Figure 23c.

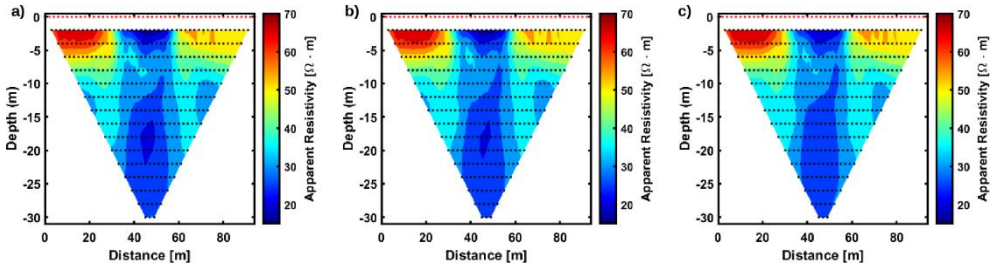


Figure 25: a) Observed pseudosection. b) Pseudosection predicted by a local least-squares inversion. c) Pseudosection computed on the solution provided by the CNN inversion.

DISCUSSION

On the one hand, deriving accurate model estimations with a limited computational effort is one of the crucial requirements of ERT inversion, and thus deterministic methods are often preferred to probabilistic inversion approaches. On the other hand, MCMC algorithms provide accurate uncertainty appraisals but their applicability to ERT is often hampered by the considerable number of forward evaluations needed to attain stable posterior estimations. In this context, the popularity of machine learning approaches has motivated us to develop an alternative inversion strategy characterized by a modest computational demand while being robust and also capable to provide quantifications of the uncertainties affecting the final solution. Our efforts were particularly devoted to combining the regression capability of CNN and the compression ability of DCT. The use of DCT reduces the dimensionality of the input and the output of the network, thereby reducing the complexity of the CNN architecture as well as the computational cost of the training phase. The choice of the number of DCT coefficients to approximate the data and the model should always constitute a compromise between the desired spatial resolution, and the dimensionality reduction of the parameter space. However, our examples showed that such a threshold level can be accurately determined from the resistivity models and the associated data drawn from the prior assumptions.

The implemented method does not require the regularization in its common-sense meaning (i.e. inclusion of model constraints into the error function). Instead, the network is trained on a data set

containing realistic subsurface scenarios and thus it learns how to reproduce a similar model that fits the input data. The main advantage of the CNN-based inversion is its high computational efficiency. Indeed, for all the examples discussed in this paper (CNN inversion, MCMC inversion, local inversion), we used Matlab codes running on a common notebook equipped with a quad-core intel Core i-7 7700HQ CPU@2.80 GHz with 16 Gb RAM.

The implemented inversion consists of four stages: data and model generation, learning process, model prediction from a given input data, and Monte Carlo simulation for uncertainty quantification. Different geostatistical simulation algorithms can be used in the first stage to generate resistivity models according to the prior assumptions. This means that the method can be used with any analytical or parametric prior distribution and spatial correlation pattern provided that appropriate geostatistical simulation codes are available to generate the training and validation examples. For example, the assumption of a log-Gaussian stationary prior model in the synthetic examples allowed us to use the popular simple kriging method, but a more sophisticated approach (i.e., the Direct Sequential Simulation method; Soares 2001) was employed in the real data application in which we assumed a non-parametric prior.

The first stage of data generation is the most computationally demanding even though it is perfectly parallelizable. The computing time for generating the ensemble of 20000 training examples was almost 6 hours for the synthetic experiments. The 10 learning processes related to the different network configurations shown in Figure 9 run in 20 minutes, approximately, whereas less than 2 minutes are requested for training the selected network configuration. These very limited computing times are guaranteed by the DCT compression of both data and model spaces and allow a fast evaluation of the performances of CNNs with different architectures and parameter settings, thereby reducing the human effort required for the network configuration. The third phase of CNN inversion gives predictions in real-time. The MC approach for uncertainty propagation can be easily parallelized and it took only 60 seconds to generate the 10000 realizations used to compute the standard deviation shown in Figure 11e. Just for comparison, a single MCMC inversion running in

761 the reduced DCT space (see Figures 11c and 11f) takes almost ten hours. This computing time would
762 dramatically grow for an MCMC inversion running in the un-reduced model space. Just for
763 comparison, a gradient-based inversion runs in two minutes, approximately, on the same hardware
764 previously mentioned.

765 We also point out that the generation of the training examples is an embarrassingly parallel
766 problem and its computing time can be dramatically reduced if a multi-core parallel architecture is
767 employed instead of a common notebook. In addition, all the codes employed in this work were
768 written in Matlab. Therefore, there is still room for a substantial decrease in the computational cost
769 related to the data generation stage, for example by employing more efficient and scalable codes
770 written in a lower-level programming language. Figure 9 also demonstrated that similar results can
771 be obtained by employing only 10000 models for the training phase. This possibility halves the
772 computing time of the data generation process. The CNN method can be also extended to 3D
773 applications but in this case, faster forward modeling routines are needed to make the computing time
774 of the generation phase affordable. Also note that the CNN inversion runs for any type of acquisition
775 layout, provided that an appropriate forward modeling code is available to compute the apparent
776 resistivity values from the subsurface resistivity model.

777 As expected, our experiments pointed out that the quality of the predictions decreases if the
778 statistical properties of the actual subsurface resistivity model and noise statics differ from those
779 assumed in the learning phase. On the one hand, we have demonstrated that the CNN inversion is
780 more sensitive to errors in the assumed prior resistivity model than to erroneous assumptions about
781 the error statistic. Moreover, we get reasonable results in most cases and only extremely wrong
782 assumptions result in meaningless predictions. This demonstrates that the proposed CNN inversion
783 is quite robust against realistic errors in the assumed noise and model properties.

784 Transfer learning can be employed to update the internal network weights when the statistical
785 properties of the target differ from those of the training examples, thus avoiding retraining the
786 network from scratch when it is applied to different datasets. In the field data experiment, the

generation of the 550 examples for transfer learning took about 10 minutes, whereas less than a minute was needed for updating the network. However, even though transfer learning can be applied to adjust the network weights when the target properties differ from those of the training examples, an accurate estimation of the prior is always crucial to obtain a new training set that exhaustively captures the target properties.

We finally point out that previous geological interpretations, borehole data, or other inversion results can be used in many applications to define the prior assumption because all such information gives a glimpse into the expected distribution of the resistivity values. The proposed approach could constitute a possible alternative to deterministic gradient-based approaches. For example, the CNN inversion can be particularly useful for monitoring applications. In this context, the vast amount of observed data, inverted models, and geologic information usually available can be exploited to derive a complete training set. The main benefit of the proposed approach over deterministic and MCMC inversions is that once the CNN is fully trained, it infers the resistivity values and the associated uncertainties from the monitoring dataset in real-time.

A possible improvement could be replacing CNNs with residual neural networks (RNNs). Compared to standard CNNs, RNNs avoid the vanishing gradient problem and allow training a deeper network that potentially can better approximate a non-linear function linking the input and the output responses. We are now working on this research topic.

CONCLUSIONS

We proposed a CNN inversion that presents a possible alternative to deterministic or probabilistic ERT inversion approaches. Instead of minimizing an error function or sampling from a posterior probability density, the proposed approach employs a trained CNN to infer the subsurface resistivity model from the apparent resistivity pseudosection. We used a Discrete Cosine Transform (DCT) reparameterization of data and model spaces to reduce the computational effort of the training phase. In our case, the DCT constitutes an additional feature extraction tool that uses orthogonal basis

functions to compress the dimensionality of the input and output of the CNN. On the other hand, the DCT reparameterization also acts as a model regularization strategy that reduces the number of unknown parameters to be estimated. The implemented inversion also incorporates a Monte Carlo approach to properly propagate onto the CNN predictions both the noise affecting the observed data and the modeling error associated with the network approximation. The CNN inversion requires a relatively small data set for training and allows us to estimate the resistivity distribution and the associated uncertainty with high precision and orders of magnitude faster than MCMC algorithms.

Our synthetic and real data experiments showed very promising results and demonstrated that a convolutional neural network can effectively approximate the inverse of a nonlinear operator that is very difficult and expensive to be computed analytically. The CNN+MC approach gives final model and data predictions comparable to those yielded by a deterministic inversion, but also provides quite accurate posterior model uncertainties that are similar to that estimated by a much more computationally demanding MCMC sampling. Our tests demonstrated that the CNN inversion is more sensitive to errors in the assumed statistical properties of the actual resistivity values rather than to errors in the assumed noise properties. The CNN achieves satisfactory predictions unless extremely wrong prior model statistics are employed for the training phase. However, transfer learning can be conveniently employed to rapidly update the internal network weights when the target and the training exhibit different statistical properties.

ACKNOWLEDGMENTS

This research was partially funded by Ministero dell'Ambiente e della Tutela del Territorio e del Mare, Italy, project DILEMMA – Imaging, Modeling, Monitoring and Design of Earthen Levees. We are grateful to LSI-Lastem S.r.l. that designed and installed the G.Re.T.A. (Geo REsistivimeter for Time-lapse Analysis) system to monitor the Colorno pilot site.

Conflict of interest

The authors declare no conflict of interest

Data Availability Statement

Data available on request from the authors

REFERENCES

Aleardi, M., and Salusti, A. (2020). Markov chain Monte Carlo algorithms for target-oriented and interval-oriented amplitude versus angle inversions with non-parametric priors and non-linear forward modellings. Geophysical Prospecting, 68(3), 735-760.

Aleardi, M. (2019). Using orthogonal Legendre polynomials to parameterize global geophysical optimizations: Applications to seismic-petrophysical inversion and 1D elastic full-waveform inversion. Geophysical Prospecting, 67(2), 331-348.

Aleardi, M., Ciabarri, F., and Gukov, T. (2018). A two-step inversion approach for seismic-reservoir characterization and a comparison with a single-loop Markov-chain Monte Carlo algorithm. Geophysics, 83(3), R227-R244.

Aleardi, M. (2020a). Combining discrete cosine transform and convolutional neural networks to speed up the Hamiltonian Monte Carlo inversion of pre-stack seismic data. Geophysical Prospecting, *68*(9), 2738-2761.

Aleardi, M., Vinciguerra, A., and Hojat, A. (2021). A geostatistical Markov chain Monte Carlo inversion algorithm for electrical resistivity tomography. Near Surface Geophysics, 19(1), 7-26.

Aleardi, M., Salusti, A., and Pierini, S. (2020b). Transdimensional and Hamiltonian Monte Carlo inversions of Rayleigh-wave dispersion curves: A comparison on synthetic datasets. Near Surface Geophysics. In print. doi: 10.1002/nsg.12100.

Aleardi, M. (2020b). Discrete cosine transform for parameter space reduction in linear and non-linear AVA inversions. Journal of Applied Geophysics, 179, 104106.

867 Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T. (2018). Deep-learning tomography. The
868 Leading Edge, 37(1), 58-66.

869 Arosio, D., Munda, S., Tresoldi, G., Papini, M., Longoni, L., and Zanzi, L. (2017). A customized
870 resistivity system for monitoring saturation and seepage in earthen levees: installation and validation.
871 Open Geosciences, 9(1), 457-467.

872 Azevedo, L., and Soares, A. (2017). Geostatistical methods for reservoir geophysics. Berlin:
873 Springer.

874 Crawford, M. M., Bryson, L. S., Woolery, E. W., and Wang, Z. (2018). Using 2-D electrical
875 resistivity imaging for joint geophysical and geotechnical characterization of shallow landslides.
876 Journal of Applied Geophysics, 157, 37-46.

877 Dahlin, T., and Zhou, B. (2004). A numerical comparison of 2D resistivity imaging with 10
878 electrode arrays. Geophysical Prospecting, 52, 379-398.

879 Das, V., Pollack, A., Wollner, U., and Mukerji, T. (2019). Convolutional neural network for
880 seismic impedance inversion. Geophysics, 84(6), R869-R880.

881 Dejtrakulwong, P., Mukerji, T., and Mavko, G. (2012). Using kernel principal component analysis
882 to interpret seismic signatures of thin shaly-sand reservoirs. In SEG Technical Program Expanded
883 Abstracts 2012 (pp. 1-5). Society of Exploration Geophysicists.

884 Fernández-Martínez, J. L., Fernández-Muñiz, Z., Pallero, J. L. G., and Bonvalot, S. (2017). Linear
885 geophysical inversion via the discrete cosine pseudo-inverse: application to potential fields.
886 Geophysical Prospecting, 65, 94-111.

887 Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). Deep learning. Cambridge: MIT
888 press.

889 Grana, D., Passos de Figueiredo, L., and Azevedo, L. (2019). Uncertainty quantification in
890 Bayesian inverse problems with model and data dimension reduction. Geophysics, 84(6), M15-M24.

891 Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. Cambridge, MA: MIT Press.

892 Hansen, T. M., and Cordua, K. S. (2017). Efficient Monte Carlo sampling of inverse problems
893 using a neural network-based forward—applied to GPR crosshole traveltime inversion. Geophysical
894 Journal International, 211(3), 1524-1533.

895 He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-
896 level performance on imagenet classification. In Proceedings of the IEEE international conference on
897 computer vision, 1026-1034.

898 Hermans, T., and Paepen, M. (2020). Combined inversion of land and marine electrical resistivity
899 tomography for submarine groundwater discharge and saltwater intrusion characterization.
900 Geophysical Research Letters, 47(3), e2019GL085877.

901 Hojat, A., Arosio, D., Ivanov, V. I., Longoni, L., Papini, M., Scaioni, M., Tresoldi, G., and Zanzi,
902 L. (2019a). Geoelectrical characterization and monitoring of slopes on a rainfall-triggered landslide
903 simulator. Journal of Applied Geophysics, 170, 103844.

904 Hojat, A., Arosio, D., Longoni, L., Papini, M., Tresoldi, G., and Zanzi, L. (2019b). Installation
905 and validation of a customized resistivity system for permanent monitoring of a river embankment.
906 In EAGE-GSM 2nd Asia Pacific Meeting on Near Surface Geoscience and Engineering, 2019(1), 1-
907 5.

Hojat A., Arosio D., Loke M.H., Longoni L., Papini M., Tresoldi G. and Zanzi L. (2019c). Assessment of 3D geometry effects on 2D ERT data of a permanent monitoring system along a river embankment. EAGE-GSM 2nd Asia Pacific Meeting on Near Surface Geoscience and Engineering, Kuala Lumpur, Malaysia, DOI: 10.3997/2214-4609.201900427.

Hojat, A., Arosio, D., Ivanov, V.I., Loke, M.H., Longoni, L., Papini, M., Tresoldi, G. and Zanzi, L. (2020). Quantifying seasonal 3D effects for a permanent electrical resistivity tomography (ERT) monitoring system along the embankment of an irrigation canal. Near Surface Geophysic, DOI: 10.1002/nsg.12110.

Huot, F., Biondi, B., and Beroza, G. (2018). Jump-starting neural network training for seismic problems. In SEG Technical Program Expanded Abstracts, Society of Exploration Geophysicist, 2191-2195.

Karaoulis, M., Revil, A., Tsourlos, P., Werkema, D. D., and Minsley, B. J. (2013). IP4DI: A software for time-lapse 2D/3D DC-resistivity and induced polarization tomography. Computers & Geosciences, 54, 164-170.

Karaoulis, M., Tsourlos, P., Kim, J. H., and Revil, A. (2014). 4D time-lapse ERT inversion: introducing combined time and space constraints. Near Surface Geophysics, 12(1), 25-34.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097-1105.

Legaz, A., Vandemeulebrouck, J., Revil, A., Kemna, A., Hurst, A. W., Reeves, R., and Papasin, R. (2009). A case study of resistivity and self-potential signatures of hydrothermal instabilities, Inferno Crater Lake, Waimangu, New Zealand. Geophysical Research Letters, 36(12).

Lewis, W., and Vigh, D. (2017). Deep learning prior models from seismic images for full-waveform inversion. 87th Annual International Meeting, SEG, Expanded Abstracts, 1512–1517.

931 Li, S., Liu, B., Ren, Y., Chen, Y., Yang, S., Wang, Y., and Jiang, P. (2019). Deep learning
932 inversion of seismic data. IEEE Transactions on Geoscience and Remote Sensing, 58(3), 2135–2149.

933 Liu, B., Guo, Q., Li, S., Liu, B., Ren, Y., Pang, Y., Liu, L., Jiang, P., (2020). Deep Learning
934 Inversion of Electrical Resistivity Data. IEEE Transactions on Geoscience and Remote Sensing,
935 58(8), 5715-5728.

936 Lochbühler, T., Breen, S. J., Detwiler, R. L., Vrugt, J. A., and Linde, N. (2014). Probabilistic
937 electrical resistivity tomography of a CO2 sequestration analog. Journal of Applied Geophysics, 107,
938 80-92.

939 Loke, M. H. (2018). Rapid 2-D Resistivity and IP inversion using the least-squares method.
940 www.geotomosoft.com, last accessed on 3 July 2020.

941 Masters, D., and Luschi, C. (2018). Revisiting small batch training for deep neural networks. arXiv
942 preprint arXiv:1804.07612.

943 Moghadas, D. (2020). One-dimensional deep learning inversion of electromagnetic induction data
944 using convolutional neural network. Geophysical Journal International, 222(1), 247-259.

945 Monajemi, H., Donoho, D.L., and Stodden, V. (2016). Making massive computational
946 experiments painless: 2016 IEEE International Conference on Big Data, 2368–2373.

947 Moradipour, M., Ranjbar, H., Hojat, A., Karimi-Nasab, S. and Daneshpajouh, S., (2016)
948 Laboratory and field measurements of electrical resistivity to study heap leaching pad no. 3 at
949 Sarcheshmeh copper mine. 22nd European Meeting of Environmental and Engineering Geophysics,
950 DOI: 10.3997/2214-4609.201602140.
951

Müller, K., Vanderborght, J., Englert, A., Kemna, A., Huisman, J. A., Rings, J., and Vereecken, H. (2010). Imaging and characterization of solute transport during two tracer tests in a shallow aquifer using electrical resistivity tomography and multilevel groundwater samplers. Water Resources Research, 46(3).

Park, M. J., and Sacchi, M. D. (2020). Automatic velocity analysis using Convolutional Neural Network and Transfer learning. Geophysics, 85(1), V33-V43.

Parzen, E. (1962). On estimation of a probability density function and mode. The annals of mathematical statistics, 33(3), 1065-1076.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.

Pidlisecky, A., and Knight, R. (2008). FW2_5D: A MATLAB 2.5-D electrical resistivity modeling code. Computers & Geosciences, 34(12), 1645-1654.

Pollock, D., and Cirpka, O. A. (2012). Fully coupled hydrogeophysical inversion of a laboratory salt tracer experiment monitored by electrical resistivity tomography. Water Resources Research, 48(1).

Pradhan, A., and Mukerji, T. (2020). Seismic Bayesian evidential learning: Estimation and uncertainty quantification of sub-resolution reservoir properties. Computational Geosciences, 1-20.

Puzyrev, V. (2019). Deep learning electromagnetic inversion with convolutional neural networks. Geophysical Journal International, 218(2), 817-832.

Ramirez, A.L., Nitao, J.J., Hanley, W.G., Aines, R., Glaser, R.E., Sengupta, S.K., Dyer, K.M., Hickling, T.L., Daily, W.D. (2005). Stochastic inversion of electrical resistivity changes using a Markov Chain Monte Carlo approach. Journal of Geophysical Research: Solid Earth, 110(B2).

Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. arXiv preprint arXiv:1801.07232.

Sambridge, M., and Mosegaard, K. (2002). Monte Carlo methods in geophysical inverse problems. Reviews of Geophysics, 40(3), 3-1.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization?. In Advances in Neural Information Processing Systems, 2483-2493.

Scherer, D., Müller, A. and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. International Conference on Artificial Neural Networks, Springer, Lecture Notes in Computer Science 6354, 92–101.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Soares, A. (2001). Direct sequential simulation and cosimulation. Mathematical Geology, 33(8), 911-926.

Szabó, N. P., and Dobróka, M. (2019). Series expansion-based genetic inversion of wireline logging data. Mathematical Geosciences, 51(6), 811-835.

Tresoldi, G., Arosio, D., Hojat, A., Longoni, L., Papini, M., and Zanzi, L. (2019). Long-term hydrogeophysical monitoring of the internal conditions of river levees. Engineering Geology, 259, 105139.

Vinciguerra, A., Aleardi, M., Hojat, A., and Stucchi E. (2020). Discrete cosine transform for parameter space reduction in Bayesian ERT inversion. Submitted for the 26th EAGE Near Surface conference 2020.

Waldeland, A. U., Jensen, A. C., Gelius, L. J., and Solberg, A. H. S. (2018). Convolutional neural networks for automated seismic interpretation. The Leading Edge, 37(7), 529-537.

Wang, B., Zhang, N., Lu, W., and Wang, J. (2019). Deep-learning-based seismic data interpolation: A preliminary result. Geophysics, 84(1), V11-V20.

Xiong, W., Ji, X., Ma, Y., Wang, Y., AlBinHassan, N. M., Ali, M. N., and Luo, Y. (2018). Seismic fault detection with convolutional neural network. Geophysics, 83(5), O97-O103.