Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Electrical Impedance Tomography Image Reconstruction Based on Neural Networks

André Bianchessi *,1 **Rodrigo H. Akamine** *,2
Guilherme C. Duran *,3 **Naser Tanabi** *,4 **André K. Sato** *,5
Thiago C. Martins *,6 **Marcos S. G. Tsuzuki** *,7

* *Laboratory of Computational Geometry*
Mechatronics and Mechanical Systems Engineering Department,
Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil

Abstract: Electrical impedance tomography (EIT) is an imaging technique with a promising future. Several methods have been used for EIT image reconstruction, such as Simulated Annealing, Gauss Newton, Kalman filter and D-Bar. Recently, some authors solved this problem using artificial neural network (ANN) through pixel by pixel reconstruction, considering a fixed resolution for the final image. This work proposes a reconstruction based on the EIT forward problem. Two different meshes were considered: a coarse and a refined mesh. The latter was used to produce simulated potentials, which are the inputs for ANN training. The nodes conductivities, which used to create the outputs for training, defined in the coarser mesh. Therefore, the proposed method consists of training the ANN with inputs from a refined mesh and outputs from a coarse mesh. Two ANN architectures are proposed and compared: one based on the LeNet architecture, and another based on the feed-forward fully connected ANN. The obtained image is not dependent on any image resolution. The preliminary results show that the LeNet architecture has better performance. *Copyright ©2020 IFAC.*

Keywords: Neural networks, Electrical impedance tomography, Computer-aided tomography.

1. INTRODUCTION

Electrical Impedance Tomography (EIT) is a novel, promising, non-invasive method for obtaining conductivity images of the interior of an object, from the electrical potential measurements taken at the object boundary. EIT is being largely used for dynamic, real-time monitoring of industrial processes and biomedical analysis (Martins and Tsuzuki, 2012; Silva et al., 2017).

In a typical EIT configuration, the electrodes are positioned at specific intervals along the contour of an object. Then, a low amplitude current pattern is applied to the object through the electrodes and the electric potentials are measured at all electrodes. Just as an illustration, Fig. 1 shows two examples of current injection through a pair of electrodes. For each injection, the electrical potential V_i at each electrode i is measured relative to a grounded electrode.

Several methods have been used for EIT image reconstruction: simulated annealing (Martins and Tsuzuki, 2015; Martins et al., 2014), Gauss Newton (Camargo, 2013), Kalman filter (Moura, 2013), D-Bar (Alsaker and Mueller, 2018). A good review on the different methods is available (Martins et al., 2019). The great majority of the available methods, excluding the D-Bar method, solves the inverse problem defined by the EIT, by iteratively evaluating the forward problem. The Neural Network approach, similarly to the D-Bar, directly determines the image reconstruction without any evaluation of the forward problem.

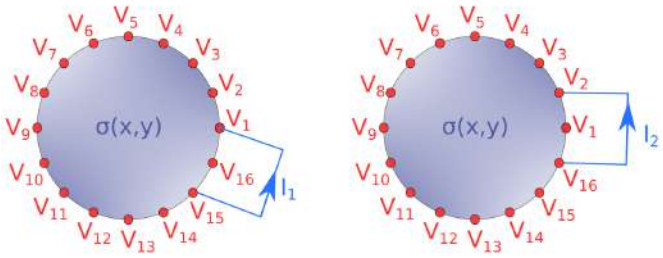

Fig. 1. Examples of current patterns applied to a cylindrical domain. Electric potentials are measured at the electrodes. Both domains have conductivity distribution $\sigma(x, y)$.

The forward EIT problem is the determination of the potential distribution in the domain, given the conductivity distribution $\sigma(x, y)$ and the injected current I_n. The fastest approach to solve the forward problem is using GPGPU (Tavares et al., 2019). The forward problem is used to train the neural network and, therefore, it is

[1] e-mail: andrebianchessi@gmail.com.
[2] e-mail: rodrigo.h.akamine@usp.br.
[3] e-mail: guiduran@usp.br.
[4] e-mail: n.tanabi@gmail.com.
[5] e-mail: andre.kubagawa@gmail.com.
[6] e-mail: thiago@usp.br.
[7] e-mail: mtsuzuki@usp.br.

Copyright lies with the authors

necessary to evaluate the forward problem many times over. Although the speed up of the training phase does not impact the performance of the solution, such improvement is not negligible as it helps the development phase.

The inverse problem solution is affected by mesh deficiencies. Hence, the image should be reconstructed using meshes with higher densities, such that errors due to the discretization are minimized. Martins et al. (2019) showed that a correction error can be used to overcome errors originated in the discretization, based on a theory described n (Kaipio and Somersalo, 2004). In this work, the proposed strategy consists of evaluating the forward problems with refined meshes to train the artificial neural network (ANN). The ANN will then be trained with negligible discretization error data. However, the evaluated EIT image will be reconstructed using a coarse mesh. Once an ANN has been trained, the solution of an inverse problem can be obtained in minimal time. Thus, ANN may be a faster alternative for practical applications.

Some authors already employed ANNs to solve the EIT inverse problem. Rymarczyk et al. (2019) used three distinct configurations of feed-forward fully connected ANN (FFFC-ANN): 1. 96-10-1 (96 predictors, 10 hidden neurons and 1 answer); 2. 96-10-10 (96 predictors, 10 hidden neurons and e 10 answers); and, 3. 96-20-10 (96 predictors, 20 hidden neurons and 10 answers). Each of these configurations represent one pixel in the reconstructed image. The EIT experiment consisted of 6 sets of measurements on a 16 electrodes configuration. Li et al. (2017) considered a topology with two stacked auto-encoders and a logistic regression. The proposals presented in (Michalikova et al., 2014; Wang et al., 2009) used an ANN with radial basis function (RBF), similar to the one presented in (Rymarczyk et al., 2019). However, in the former case the image is completely reconstructed using a unique ANN (928 predictors, 100 hidden neurons and 3.214 answers). Tan et al. (2018) used the LeNet ANN (Lecun et al., 1998) to solve the EIT problem.

Previous approaches reconstructed the EIT image at the pixel level. A different approach is adopted herein. The EIT image is reconstructed considering a coarse mesh and the ANN evaluates the conductivity at the mesh nodes. The final image can be zoomed in or out, as the resolution of the image has no influence. Additionally, the ANN is trained using a refined meshed such that it is not necessary to consider the discretizaton error.

This paper has the following structure. Section 2 describes the finite element formulation for the forward problem. Section 3 describes two different proposed topologies of ANN to solve the EIT problem. The first topology is based on the LeNet architecture and the second topology is a FFFC-ANN. Section 4 has the results and section 5 has the conclusions and future works.

2. FINITE ELEMENT FORMULATION FOR THE FORWARD PROBLEM

The forward problem determines the potential distribution in the domain given the conductivity distribution and current pattern applied to the electrodes as inputs. Due to its ability to model arbitrary geometries and various

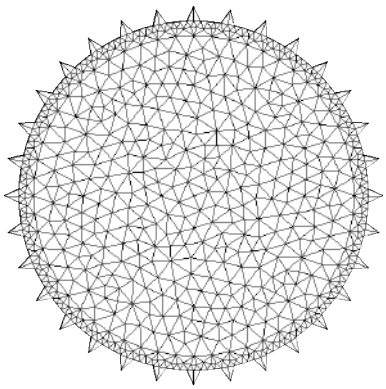

Fig. 2. Mesh example for the FEM with two regions: external ring and internal circle.

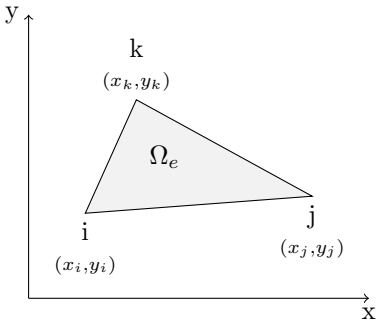

Fig. 3. Triangular element with nodes i, j and k.

boundary conditions, the finite element method (FEM) is the most useful method currently used for the numerical solution of EIT forward problems (Dong et al., 2005). For the bidimensional case, the problem must comply with the Laplace equation

$$\nabla \cdot [\sigma(x,y) \cdot \nabla\phi(x,y)] = 0 \qquad (x,y) \in \Omega \qquad (1)$$

where σ and ϕ are the conductivity and potential distributions, respectively, and Ω is the domain region. The zero result indicates that there are no current sources inside the domain (Martins and Tsuzuki, 2015; Martins et al., 2014).

In the FEM, the domain is discretized, generating a mesh of smaller elements. A mesh with two regions is proposed: external ring and internal circle (see Fig. 2). The definition of two regions allows the use of distinct mesh densities. The external ring is directly connected to the electrodes. If more elements are desired in the external ring then the electrode model must allow variable number of elements as well. Both the internal circle and the electrodes have independent and controllable refinement levels. The external ring is a transition region between the electrode and the internal circle (Tavares et al., 2014).

As the linear triangular element was chosen for the finite element model, a linear interpolation polynomial is adopted to determine the potential inside the element. Thus, using the notation shown in Fig. 3, the potential inside the domain Ω_e is given by

$$\phi(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \qquad (2)$$

where α_1, α_2 and α_3 are unknown constants to be determined. Therefore, the electrical potential on nodes i, j and k of the triangular element are given by

$$\begin{aligned}\phi_i(x,y) &= \phi(x_i,y_i) = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \\ \phi_j(x,y) &= \phi(x_j,y_j) = \alpha_1 + \alpha_2 x_j + \alpha_3 y_j \\ \phi_k(x,y) &= \phi(x_k,y_k) = \alpha_1 + \alpha_2 x_k + \alpha_3 y_k\end{aligned} \tag{3}$$

Thus follows that

$$\begin{bmatrix}\phi_i \\ \phi_j \\ \phi_k\end{bmatrix}^T = \begin{bmatrix}\alpha_i \\ \alpha_j \\ \alpha_k\end{bmatrix}^T \cdot \begin{bmatrix}1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k\end{bmatrix} \tag{4}$$

The solution for the α parameters is

$$\begin{aligned}\alpha_i &= \frac{1}{2S}(a_i\phi_i + a_j\phi_j + a_k\phi_k) \\ \alpha_j &= \frac{1}{2S}(b_i\phi_i + b_j\phi_j + b_k\phi_k) \\ \alpha_k &= \frac{1}{2S}(c_i\phi_i + c_j\phi_j + c_k\phi_k)\end{aligned} \tag{5}$$

where S is the area of the triangle and

$$\begin{aligned}&a_i = x_j y_k - x_k y_j \;\; a_j = x_k y_i - x_i y_k \;\; a_k = x_i y_j - x_i y_j \\ &b_i = y_j - y_k \;\; b_j = y_k - y_i \;\; b_k = y_i - y_j \\ &c_i = x_k - x_j \;\; c_j = x_i - x_k \;\; c_k = x_j - x_i.\end{aligned} \tag{6}$$

Therefore, eq. (4) can be written as

$$\phi(x,y) = \begin{bmatrix}N_i^e & N_j^e & N_k^e\end{bmatrix} \begin{bmatrix}\phi_i \\ \phi_j \\ \phi_k\end{bmatrix} = [N]_e[\phi]_e \tag{7}$$

where

$$\begin{aligned}N_i^2 &= \frac{1}{2S}(a_i + b_i x + c_i x) \\ N_j^2 &= \frac{1}{2S}(a_j + b_j x + c_j x) \\ N_k^2 &= \frac{1}{2S}(a_k + b_k x + c_k x)\end{aligned} \tag{8}$$

The partial derivation of $\phi(x,y)$ is given by

$$[\nabla\phi] = \begin{bmatrix}\partial\phi/\partial x \\ \partial\phi/\partial y\end{bmatrix} = \frac{1}{2S}\begin{bmatrix}b_i & b_j & b_k \\ c_i & c_j & c_k\end{bmatrix}\begin{bmatrix}\phi_i \\ \phi_j \\ \phi_k\end{bmatrix} \tag{9}$$

By setting

$$[BC]_e = \frac{1}{2S}\begin{bmatrix}b_i & b_j & b_k \\ c_i & c_j & c_k\end{bmatrix}_e, \tag{10}$$

the Laplace equation for the equivalent variational problem can be described as

$$F(\phi) = \int_s \frac{1}{2}\sigma\left[\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2\right] dxdy \tag{11}$$

Then, combining eq. (9) with eq. (11) results in

$$\begin{aligned}F_e(\phi) &= \int_{S_e} \frac{1}{2}\sigma[\nabla\phi]^T[\nabla\phi]\,dxdy \\ &= \frac{1}{2}[\phi]_e^T\left\{\sigma\int_{S_e}[BC]_e^T[BC]_e dxdy\right\}[\phi]_e \\ &= \frac{1}{2}[\phi]_e^T[K]_e[\phi]_e\end{aligned} \tag{12}$$

where

$$\begin{aligned}[K]_e =& \sigma[BC]_e^T[BC]_e\int_{S_e} dxdy \\ =& \frac{\sigma}{4S}\begin{bmatrix}b_i & c_i \\ b_j & c_j \\ b_k & c_k\end{bmatrix}\begin{bmatrix}b_i & b_j & b_k \\ c_i & c_j & c_k\end{bmatrix} \\ =& \frac{\sigma}{4S}\begin{bmatrix}b_i^2 + c_i^2 & b_ib_j + c_ic_j & b_ib_k + c_ic_k \\ b_jb_i + c_jc_i & b_j^2 + c_j^2 & b_jb_k + c_jc_k \\ b_kb_i + c_kc_k & b_kb_j + c_kc_j & b_k^2 + c_k^2\end{bmatrix}.\end{aligned} \tag{13}$$

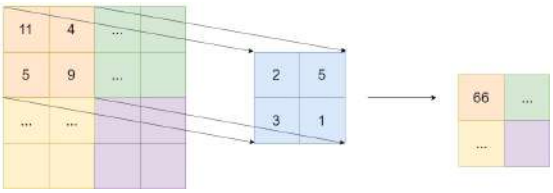

Fig. 4. Convolution example, where filters are applied to identify input features.

The element matrix $[K]_e$ must be determined for all mesh elements. Then, they are gathered according to the node numbers to generate the global conductivity matrix K. The FEM formulation for the EIT forward problem is then given by

$$K \cdot \phi = I \tag{14}$$

where I is the current vector and ϕ is the potentials vector, which are unknowns. Both vectors have dimension of n, which is the number of nodes. The matrix K is $n \times n$ and is symmetric and sparse. Therefore, the forward problem is converted into the solution of a linear system of equations.

3. PROPOSED METHODS USING ANN'S

In this section, two different topologies to solve the EIT problem are proposed for ANNs. The first proposed method is based on the LeNet architecture (Lecun et al., 1998) and the second proposed method is a FFFC-ANN with one hidden layer. Both approaches were trained using forward problems evaluated with refined meshes, and the outputs are conductivities at the coarse mesh nodes.

3.1 Convolutional ANN

A convolutional ANN consists of 3 steps: convolution, activation function and pooling. Afterwards, it is possible to pass it to a fully connected layer, as in the LeNet architecture (Lecun et al., 1998).

In the convolution step, filters (kernels) that learn to identify characteristics of the input are applied. The convolution consists of multiplying a matrix of weights $w_{i,j}$ of size $m \times n$, centered at the element (p,q) of the original matrix, with each value $x_{i,j}$ of the input where they overlap. Then, all the values of the multiplication result are summed and it is assigned to the pixel $y_{p,q}$ of the resulting matrix. Therefore, the convolution process can be described by

$$y_{p,q} = \sum_{i=1}^{m}\sum_{j=1}^{n}(x_{i,j} \cdot w_{i,j})\,.$$

After the filter passed through the entire input, a matrix is generated as a result. This process is illustrated in Fig. 4.

After the convolution, an activation function called *ReLU* is applied to each value of the matrix and compared with a bias $b_{i,j}$ of the output matrix. *ReLU* stands for Rectified Linear Unit, and it is defined as

$$f(y_{i,j} = x_{i,j} \cdot w_{i,j} + b_{i,j}) = \begin{cases}y_{i,j}, & y_{i,j} \geq 0 \\ 0, & y_{i,j} < 0\end{cases}. \tag{15}$$

Finally, in the pooling step, a filter of a determined size is applied to with the resulting matrix, replacing its value

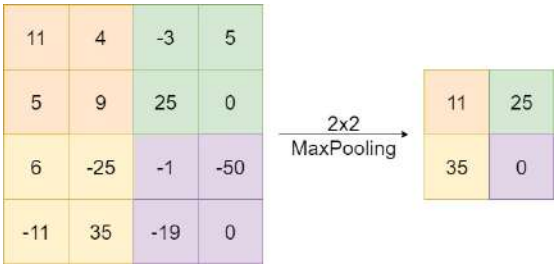

Fig. 5. MaxPooling example with a 2×2 filter.

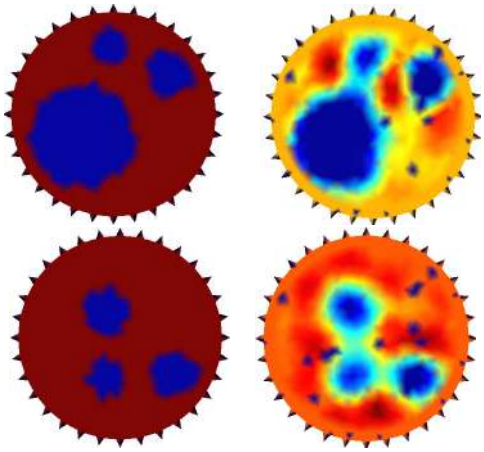

Fig. 6. Comparison between real domain (left column) and original LeNet output (right column) using the validation data set.

with the maximum value in the filter range. This method is called *MaxPooling* and it is used to prevent overfitting. An example can be seen in the Fig.5.

A voltage matrix of shape 32×32 was used as input for this architecture, and an one-dimensional array with 710 conductivity values was used as output.

The hyperparameters used were: batch size of 250 during 1000 epochs (while using Keras callbacks *EarlyStopping* to monitor the *validation loss* parameter to prevent overfitting), *Adam* optmizer and *Mean Squared Error* loss function. These parameters were used to train both the original LeNet and the modified one.

Using the original LeNet architecture, the results were not satisfactory, and this can be due to the fact that it was built originally for a classification problem, which is much simpler and has less output parameters. Therefore, its architecture needed to be modified to be more complex. The results for the original LeNet can be seen in Fig. 6.

3.2 Proposed modified LeNet architecture

The modified architecture for the convolutional neural network has 7 layers, and each of layers is described. The first layer is the input. The second layer is the convolution with 30 filters of shape 5×5, activation function $ReLU$, strides of 2×2 and padding. The third layer is the pooling with kernels of shape 2×2 and strides of 2×2. The fourth layer is the second convolution, similar to the first convolution with 60 filters. The fifth is the second pooling which is the same as the first pooling. The sixth layer flattens the data and passes onto a hidden layer with 355 neurons with activation function $ReLU$, and finally, the seventh layer, the output layer with 710 neurons and activation function $ReLU$. The proposed architecture is shown in the Fig. 7

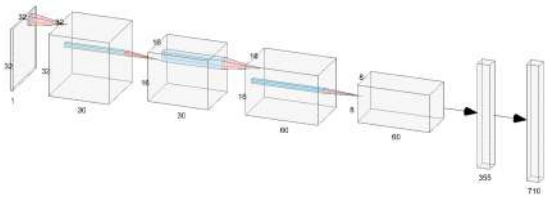

Fig. 7. Architecture of the proposed LeNet with seven layers.

3.3 Proposed FFFC-ANN architecture

The connections between FFFC-ANN nodes do not form a cycle, and every node in layer i is connected with every node in layer $i-1$ for $i > 1$, where i is the layer number starting at 1 on the first layer. For this kind of ANN, an architecture with 3 layers was adopted (1 input layer, 1 hidden layer and 1 output layer), as shown in Fig. 8.

The ANN input is an array of size 1024 that describes the potentials at the electrodes for each current pattern applied to a domain. It follows that the first and second members of the input array correspond, to the potentials at electrode 1 and 2, the resulting matrix , for the first current pattern applied. The 33rd member corresponds to the the potential at electrode 1 for the second current pattern; and so on until the last member, that corresponds to the potential at electrode 32 for the 32nd current pattern. The network output is an array with size n in which the $i-th$ value represents the electrical conductivity at node i in the coarse mesh. Following literature such as Fernández-Fuentes et al. (2018), hyperbolic tangent was used as activation function for the input layer, and Rectified Linear Unit ($ReLU$) was used for the hidden layer.

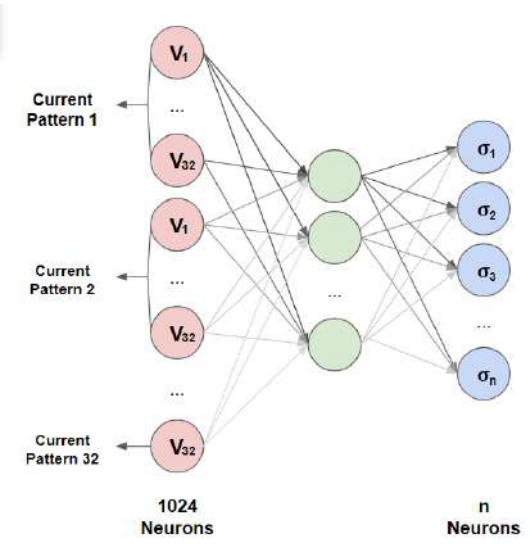

Fig. 8. Proposed FFFC-ANN architecture.

Table 1. FFFC-ANN training parameters determined by means of a random search. NHL: neurons in hidden layer

NHL	Loss Function	Batch size	Optimizer
173	MSE	107	Adam

The training parameters have been optimized by means of a random search (see Fig. 9) which aimed to find the parameters that would minimize the mean squared error (MSE) calculated on the validation data set. Although the number of epochs was kept at 1000, *Early stopping* with a patience of 20 was used during training, which would stop the training process if the MSE calculated on the validation data set did not improve for 20 consecutive epochs, and was triggered for all parameters tested. The parameters that achieved the lowest MSE, which was 0.036, are shown in Table 1.

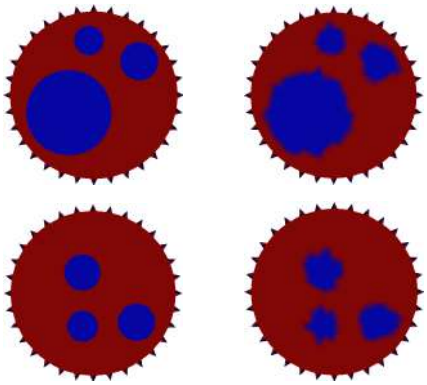

Fig. 9. Example of randomly generated conductivity distributions for a mesh with 43904 nodes (left column) and 710 nodes (right column)

4. RESULTS

Training and validation data sets were created and used to train the two proposed ANN architectures. Once trained, the ANNs were evaluated with the validation data set and tested with new data. Both ANN architectures outputs were compared.

4.1 Synthetic Training Data Generation

The training data was generated by solving the forward problem using a set of current patterns applied to a domain with 32 electrodes with many randomly generated conductivity distributions. The set of current patterns chosen consisted of a current with amplitude 1.9mA applied between electrodes i and $i+4$ for $1 \leq i \leq 28$ (skip-3). Therefore, 32 current patterns were applied to each domain generated. The algorithm used to generate training examples had the following steps:

(1) randomly define a circular domain with 32 electrodes and 3 randomly placed circular regions with random radius inside it so that the conductivity is 0.002 inside the circular regions, 0.005 at the electrode contact points and 0.3815 elsewhere;
(2) create 2 mesh files for the domain: a very refined one with $43,904$ nodes, as shown in Fig. 10, and a coarse one with 710 nodes, as shown in Fig. 2;
(3) solve the forward problem using the very refined mesh for each one of the current patterns;
(4) restart from (1) until enough data have been generated.

The final output of the algorithm is a set of mesh files with 710 nodes that describe the normalized domain conductivity distribution and a .csv files with $1,024$ lines, where the first 32 lines represent the normalized potentials, calculated with a very refined mesh, at each electrode for the first current pattern, the next 32 lines represent these potentials for the second current pattern, and so on. Fig. 2 represents the mesh geometry and Fig. 9 shows examples of normalized conductivity distributions generated for a refined and a coarse mesh.

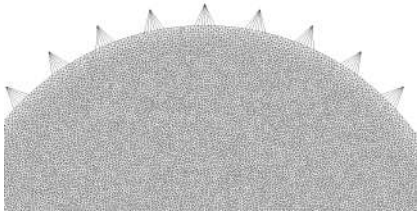

Fig. 10. Section of very refined mesh geometry with 43904 nodes

A data set with $10,004$ cases was created – $9,500$ cases were used for training both proposed ANN architectures, 500 were used as validation data during training, and the last 4 were kept aside and used for tests.

The modified LeNet and the FFFC-ANN were tested using the same data set, which consisted of 4 cases. Fig. 11 shows the real conductivity distributions from each case in the test data set and the predictions from the FFFC-ANN and LeNet ANN. Table 2 shows the absolute mean error between the real and predicted conductivity distribution, where the domain number i is corresponds to the ith row in Fig. 11.

Table 2. Absolute error between real and predicted conductivities for the proposed FFFC-ANN and LeNet ANN architectures

Domain number (row in Fig. 11)	FFFC-ANN	LeNet
1	0.089	0.052
2	0.092	0.039
3	0.088	0.057
4	0.088	0.053

5. CONCLUSION AND FUTURE WORK

Two ANN architectures were proposed to solve the EIT image reconstruction problem. The training was performed using two different meshes: a coarse mesh to obtain the conductivities at its nodes and a refined mesh to obtain the simulated potentials at the electrodes. The obtained results are not associated with any pixel resolution. As seen in Table 2, they showed that the LeNet ANN archicture has better performance when compared with the FFFC-ANN.

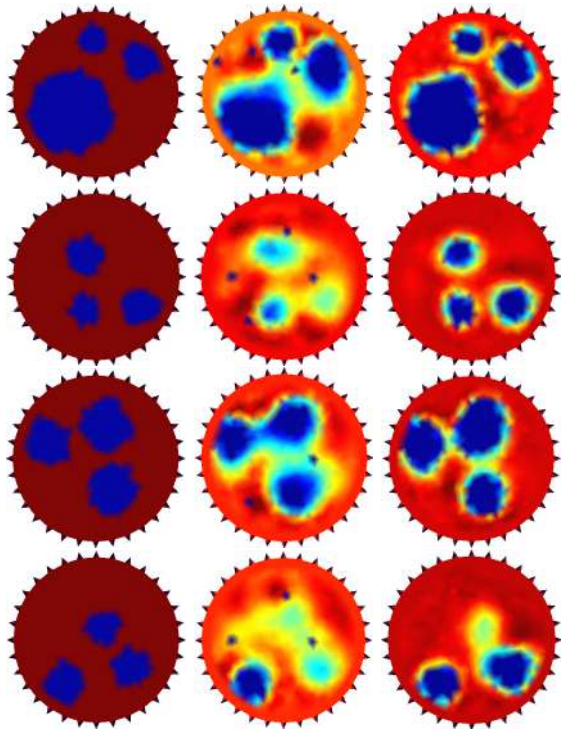

Fig. 11. Comparison between real domain (left column), FFFC-ANN output (middle column) and modified LeNet output (right column) using the test data set.

In this work, only the real part of the conductivities were considered. As future work, capacitive effects, anisotropy and three dimensionality should be contemplated by the forward problem solver. Additional research also includes electrode position and boundary geometry determination.

ACKNOWLEDGEMENTS

This project is supported by CNPq (proc. 433.151/2018–8), FAPESP (proc. 2017/07799–5) and NAP TIE-US (USP). A. Bianchessi is supported by CNPq (proc. 100.919/2019-7). R. H. Akamine is supported by FAPESP (proc. 2018/10549-3). N. Tanabi and A. K. Sato is supported by FUSP/Petrobras. M. S. G. Tsuzuki and Thiago C. Martins are partially supported by CNPq (prosc. 305.959/2016–6 and 311.195/2019-9).

REFERENCES

Alsaker, M. and Mueller, J.L. (2018). Use of an optimized spatial prior in D-bar reconstructions of EIT tank data. *Inverse Probl Imag*, 12(4), 883–901.

Camargo, E.D.L.B. (2013). *Desenvolvimento de algoritmo de imagens absolutas de Tomografia por Impedância Elétrica para uso clínico*. Phd thesis, EPUSP, São Paulo.

Dong, G., Zou, J., Bayford, R.H., Ma, X., Gao, S., Yan, W., and Ge, M. (2005). The comparison between fvm and fem for eit forward problem. *IEEE T Magn*, 41(5), 1468–1471.

Fernández-Fuentes, X., Mera, D., Gómez, A., and Vidal-Franco, I.o. (2018). Towards a fast and accurate eit inverse problem solver: A machine learning approach. *Electronics*, 7(12).

Geuzaine, C. and Remacle, J.F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Int J Numer Meth Eng*, 79(11), 1309–1331.

Kaipio, J.P. and Somersalo, E. (2004). *Statistical and computational inverse problems*, volume 160. Springer.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11), 2278–2324.

Li, X., Lu, Y., Wang, J., Dang, X., Wang, Q., Duan, X., and Sun, Y. (2017). An image reconstruction framework based on deep neural network for electrical impedance tomography. In *2017 IEEE ICIP*, 3585–3589.

Martins, T.C. and Tsuzuki, M.S.G. (2012). Electrical impedance tomography reconstruction through simulated annealing with total least square error as objective function. In *Proc 34th IEEE EMBC*, 1518–1521. San Diego, USA.

Martins, T.C. and Tsuzuki, M.S.G. (2015). EIT image regularization by a new multi-objective simulated annealing algorithm. In *Proc 37th IEEE EMBC*, 4069–4072. Milan, Italy.

Martins, T.C., Fernandes, A.V., and Tsuzuki, M.S.G. (2014). Image reconstruction by electrical impedance tomography using multi-objective simulated annealing. In *IEEE 11th ISBI*, 185–188. Beijing, China.

Martins, T.C., Sato, A.K., Moura, F.S., Camargo, E.D.L.B., Silva, O.L., Santos, T.B.R., Zhao, Z., Möeller, K., Amato, M.B.P., Mueller, J.L., Lima, R.G., and Tsuzuki, M.S.G. (2019). A review of electrical impedance tomography in lung applications: Theory and algorithms for absolute images. *Annu Rev Control*. doi: https://doi.org/10.1016/j.arcontrol.2019.05.002.

Michalikova, M., Abed, R., Prauzek, M., and Koziorek, J. (2014). Image reconstruction in electrical impedance tomography using neural network. In *2014 CIBEC*, 39–42.

Moura, F.S. (2013). *Estimação não linear de estado através do Unscented Kalman Filter na tomografia por impedância elétrica*. Phd thesis, EPUSP, São Paulo.

Rymarczyk, T., Kłosowski, G., Kozłowski, E., and Tchórzewski, P. (2019). Comparison of selected machine learning algorithms for industrial electrical tomography. *Sensors (Basel)*, 19(7), 1521.

Silva, O.L., Lima, R.G., Martins, T.C., Moura, F.S., Tavares, R.S., and Tsuzuki, M.S.G. (2017). Influence of current injection pattern and electric potential measurement strategies in electrical impedance tomography. *Control Eng Pract*, 58, 276–286.

Tan, C., Lv, S., Dong, F., and Takei, M. (2018). Image reconstruction based on convolutional neural network for electrical resistance tomography. *IEEE Sensors Journal*, 19(1), 196–204.

Tavares, R.S., Nakadaira-Filho, F.A., Tsuzuki, M.S.G., Martins, T.C., and Lima, R.G. (2014). Discretization error and the EIT forward problem. *IFAC-PapersOnLine*, 47(3), 7535–7540. 19th IFAC WC.

Tavares, R.S., Sato, A.K., Martins, T.C., Lima, R.G., and Tsuzuki, M.S.G. (2019). GPU acceleration of absolute EIT image reconstruction using simulated annealing. *Biomed Signal Proces*, 52, 445–455.

Wang, P., Li, H., Xie, L., and Sun, Y. (2009). The implementation of FEM and RBF neural network in EIT. In *2009 ICINIS*, 66–69.