

## Research Article

# Simultaneous Estimation of Wall and Object Parameters in TWR Using Deep Neural Network

Fardin Ghorbani  and Hossein Soleimani 

*School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran*

Correspondence should be addressed to Hossein Soleimani; [hsoleimani@iust.ac.ir](mailto:hsoleimani@iust.ac.ir)

Received 16 September 2022; Revised 14 December 2022; Accepted 19 December 2022; Published 28 December 2022

Academic Editor: Praveen Kumar Malik

Copyright © 2022 Fardin Ghorbani and Hossein Soleimani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of this paper is to present a deep learning model that simultaneously estimates targets and wall parameters in through-the-wall radar (TWR). As a result of the complexity of the environments in which through-the-wall radars operate, TWR faces many challenges. The propagation of radar signals through walls is further delayed and attenuated than in free space. Therefore, the targets are less able to be detected and the images of the targets are distorted and defocused as a consequence. To address the above challenges, two modes are considered in this work: single targets and two targets. In both cases, permittivity and wall thickness are considered, along with the target's center in two dimensions and the permittivity of targets. Therefore, in the case of a single target, we estimate five values, whereas in the case of two targets, we estimate eight values simultaneously, each representing the mentioned parameters. As a result of using deep neural networks to solve the task of target locating problem in TWR, the model has a better chance of learning and increased accuracy if it involves more parameters (such as wall parameters and permittivity of the wall) in the target location problem. In this way, the accuracy of target locating improved when two wall parameters were considered in problem. A deep neural network model was used to estimate wall permittivity and thickness, as well as two-dimensional coordinates and permittivity of targets with 99% accuracy in single-target and two-target modes.

## 1. Introduction

Recently, through-the-wall imaging (TWRI) has become one of the most attractive fields of research that has different applications to locate, identify, and classify different targets [1–3]. Through-the-wall radar (TWR) faces many challenges due to the complex nature of the environments in which it operates. In comparison to signal propagation in free space, radar returns passing through walls are further delayed and attenuated. This reduces the detecting abilities of the targets, and the images of the targets are distorted and defocused as a consequence [4, 5]. To overcome these challenges, different methods and techniques are employed [6–9]. The TWR challenge can also be addressed by machine learning algorithms [10, 11].

Recent advancement in machine learning algorithms, particularly deep learning, and their penetration into other sciences have solved novel problems in various fields. The

impact of machine learning on various fields demonstrated the high flexibility and ability to improve previous results and solve new problems [12–19]. These characteristics of machine learning, particularly deep learning, to discover hidden signal patterns make it an excellent tool for analyzing radar signals. Among machine learning applications in radar, we focus on TWR and investigate how it can solve new problems in this field that conventional methods cannot [20, 21].

The estimation of wall parameters and the location of targets are two important applications of TWR that are gaining much attention. In each case, there are many challenges facing each of these applications. As a result of walls and other objects, there is ambiguity and interference in the received signal in target locating [22]. Furthermore, estimating the wall parameters to locate the targets helps to provide a more complete picture of the target, which is not a straightforward process [23, 24].

Generally, there are two types of methods for estimating wall and target parameters: conventional and machine-based. Methods such as time-delay [25], filter-based methods [26], *M*-Sequence sensors, and continuous basis estimators [27] are conventional methods for estimating wall parameters. A time-delay estimation approach also requires at least two experiments, which is time-consuming. By analyzing the time delay between different antenna intervals, it estimates the parameters. In the filter-based method, once the wall effects are removed, the filters are constructed in both the echo domain and image domain, and the estimation method is used to obtain the best focusing parameters. The method can only estimate the wall's thickness and permittivity. A *M*-Sequence sensor used a metallic wall as a target and a sequence sensor was designed, and an echo time delay was calculated by compressive sensing. It is not practical to select metallic walls as target in a practical scenario, and estimation methods to estimate the thickness and permittivity of walls are complex.

Machine learning-based methods can be classified into two categories: those that utilize conventional machine learning algorithms such as SVM and methods based on deep learning.

In the estimation of wall parameters, Zhang et al. in [28, 29] attempted to estimate the wall parameters using an SVM-based method. Considering the scenarios mentioned, it is work only when target is unchanged. Since the location of the object is fixed, the generalization of the model is low, and it can only estimate the parameters of a wall when the target is fixed in a specific location. Wood et al. in [20] used machine learning methods for the reconstruction of target material properties.

Also, in the target locating, Zhang et al. [30] developed an SVM-based method for two-dimensional locating under a homogeneous wall and a circular metal cylinder object. Also, in [22], Zhang et al. presented a 3D positioning method proposed for a homogeneous wall for a spherical metallic object using an extreme learning machine. The method used can be used for one metallic target, and it has not been evaluated for purposes where the permittivity range is close to the human body and multitargets application. Wood et al. [20] investigated a machine learning (ML) approach for predicting the location of targets. This work performed two-dimensional positioning with a circular object using the *K*-Nearest Neighbors (KNN) algorithm and a homogeneous nonmagnetic wall.

In TWR, conventional methods are used to target locating and estimate wall parameters independently. In some of these methods, the wall effect must be removed to locate targets because ambiguities in the wall parameters distort the imaging and shift the target location. On the other hand, conventional methods are incapable of estimating the target characteristics and instead concentrate exclusively on the target location. It was also very time-consuming and complicated to estimate the target parameters in previous studies, and they only worked when the target position behind the wall was fixed and only one target was available. Furthermore, they are limited to single-target mode when estimating target location and properties.

In this work, we presented a model for simultaneously estimating the wall permittivity and the thickness, as well as the two-dimensional location of targets and the permittivity of targets, using a deep learning approach. In [21], we proposed two-dimensional positioning for the case that the wall is modeled as a complex electromagnetic wall by presenting three deep learning models. The wall is assumed to be a perfect nonmagnetic dielectric. We attempt to estimate the wall parameters and target parameters simultaneously using a deep neural network. Wall parameters include permittivity and thickness, and target parameters include target location and permittivity.

## 2. Methodologies

**2.1. Deep Learning.** Deep learning is a subset of machine learning and artificial intelligence that closely mimics the process by which the human mind acquires knowledge. This type of learning is critical in data science, which also encompasses statistics and forecasting modeling. Deep learning is highly beneficial for data scientists responsible for collecting, analyzing, and interpreting large amounts of data, as it speeds up and simplifies the process. Deep learning is the process of learning through neural networks with numerous hidden layers, and the deeper these layers go, the more complex and complete the models become. Deep learning is distinguished by its approach to solving problems. Working with conventional machine learning algorithms such as SVM, begins with manual feature extraction. Then, a machine learning model is constructed using these features. However, deep learning is designed so that the computer automatically detects and extracts the relevant features. Additionally, deep learning employs end-to-end learning, in which raw data is fed into the neural network and assigned a task, such as classification. Deep learning learns how to do this automatically. Deep learning models are taught using large sets of labeled data and neural network architectures that automatically learn features from the data without the need to extract them manually. A neural network is composed of multiple layers of neurons. Typically, neural networks consist of three layers: input, hidden, and output. As the number of layers and neurons in each hidden layer increases, the model becomes more complex. As the number of hidden layers and neurons in our network increases, it transforms into a deep neural network called deep learning.

Figure 1 shows an overview of artificial neurons. The inputs (input neurons) are  $X_1, X_2, \dots$ . Each  $X_i$  in a neural network has a weight, denoted by  $W_i$ . Indeed, each input is weighted independently. The neural network sum function (sigma) then adds the products of the  $X$  and  $W$ , and the activation function calculates the output value based on this calculation. The output of a neuron can be expressed as if the activation function is represented by Associate and  $b$  is the bias value. The neuron output can be described as follows:

$$Y = F\left(\sum_{i=1}^n W_i X_i + b_i\right). \quad (1)$$

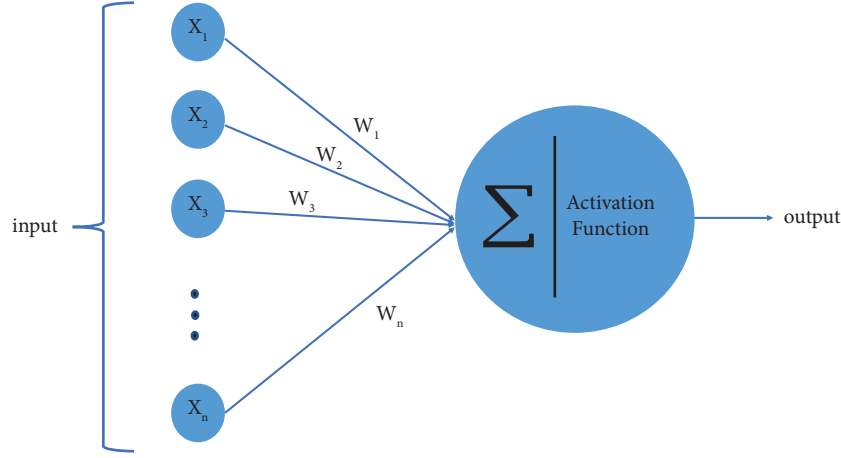


FIGURE 1: An overview of an artificial neuron.

**2.2. Transfer Learning.** Transfer learning is the application of the knowledge of a pretrained model to a different but related issue. This method allows us to train deep neural networks with less data. Learning with little data is very valuable and requires fewer hardware resources; on the other hand, in most fundamental problems, there is little data that can be used to train models. Transfer learning allows us to easily extend the model's knowledge acquired in one area to other problems. In other words, instead of starting the training from the beginning, we can use the patterns obtained in the same problem to solve the new problem more efficiently. In transfer learning, the first layer and the middle (hidden) layers in a pretrained neural network are usually kept, and the output layer in a new network becomes replaced with another layer. Then the whole network is trained again with new problem data. The main advantage of transfer learning can be summarized in two ways: first, reduced learning time and the need for fewer hardware resources, and second, a more straightforward generalization of the problem to other issues.

**2.3. FDTD.** The finite difference time-domain (FDTD) is a method for solving Maxwell's equations. The equations of Ampere's Law and Faraday's Law can be written as follows:

$$\begin{aligned} \nabla \times H &= \frac{\partial D}{\partial t} = \epsilon \frac{\partial E}{\partial t}, & (\text{Faraday's Law}), \\ \nabla \times E &= -\frac{\partial B}{\partial t} = \mu \frac{\partial H}{\partial t}, & (\text{Ampere's Law}), \end{aligned} \quad (2)$$

We use  $TM_z$  polarization to rewrite Ampere's Law and Faraday's Law as [31]

$$\epsilon \frac{\partial E}{\partial t} = \nabla \times H = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ H_x & H_y & 0 \end{vmatrix} = \hat{a}_z \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (3)$$

$$-\mu \frac{\partial H}{\partial t} = \nabla \times E = \begin{vmatrix} \hat{a}_x & \hat{a}_y & \hat{a}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & E_z \end{vmatrix} = \hat{a}_x \frac{\partial E_z}{\partial y} - \hat{a}_y \frac{\partial E_z}{\partial x}. \quad (4)$$

The scalar equations for  $TM_z$  are obtained from (3) and (4):

$$-\mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial y}, \quad (5)$$

$$\mu \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x}, \quad (6)$$

$$\epsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}. \quad (7)$$

Equations (5)–(7) can be written in finite-differences form, and future fields can be expressed in terms of past fields due to the space-time discretization. The indexes  $m$  and  $n$  denote the spatial steps in the  $x$  and  $y$  directions, respectively, and the index  $q$  corresponds to the temporal step. Additionally, the spatial step sizes are  $\Delta x$  and  $\Delta y$  in the  $x$  and  $y$  directions, respectively. The finite difference approximation of (5) expanded about the space-time point  $(m\Delta x, (n+1/2)\Delta y, q\Delta t)$ . The resulting equation is

$$-\mu \frac{H_x^{q+1/2}[m, n+1/2] - H_x^{q-1/2}[m, n+1/2]}{\Delta t} = \frac{E_z^q[m, n+1] - E_z^q[m, n]}{\Delta y}. \quad (8)$$

The equation can be rewritten as follows for future value in terms of past value:

$$H_x^{q+1/2}\left[m, n + \frac{1}{2}\right] = H_x^{q-1/2}\left[m, n + \frac{1}{2}\right] - \frac{\Delta t}{\mu \Delta y} (E_z^q[m, n+1] - E_z^q[m, n]). \quad (9)$$

We can also write for equations (6) and (7) expanded about the space-time point  $((m+1/2)\Delta x, n\Delta y, q\Delta t)$  and  $(m\Delta x, n\Delta y, (q+1/2)\Delta t)$ , respectively:

$$\begin{aligned} H_y^{q+1/2}\left[m + \frac{1}{2}, n\right] &= H_y^{q-1/2}\left[m + \frac{1}{2}, n\right] + \frac{\Delta t}{\mu \Delta x} (E_z^q[m+1, n] - E_z^q[m, n]), \\ E_z^{q+1}[m, n] &= E_z^q[m, n] + \frac{\Delta t}{\epsilon \Delta x} \left\{ H_y^{q+1/2}\left[m + \frac{1}{2}, n\right] - H_y^{q-1/2}\left[m - \frac{1}{2}, n\right] \right\} \\ &\quad - \frac{\Delta t}{\epsilon \Delta y} v \left\{ H_y^{q+1/2}\left[m, n + \frac{1}{2}\right] - H_y^{q+1/2}\left[m, n - \frac{1}{2}\right] \right\}. \end{aligned} \quad (10)$$

**2.4. Data Gathering.** The FDTD library (<https://github.com/flaport/fdtd>) in Python is used to simulate a two-dimensional TWR problem. In this case, a 30 cm square considered with  $\epsilon_r = 80$  as the target. A 3 GHz plane wave is generated using the FDTD library using a line source.  $TM_z$  polarization assumed, which implies that  $E_x = E_y = H_z = 0$  and  $E_z, H_x, H_y$  are nonzero. The source's emitted wave hits the wall, and some of it returns, while the remainder passes through the wall, hits the target, and scatters away from the target. Finally, the scattered wave is received by the detector. In this step, the fields of  $E_z, H_x, H_y$  was retrieved and used to create the required dataset. Also, a homogeneous wall with  $\mu = 1$ ,  $\sigma = 0$ , and  $\epsilon_r$  used for data generation varies from 3 to 9. To create a single-target dataset, the target moved in the two-dimensional space specified in Figure 2, also change the target permittivity from 5 to 85, the wall permittivity from 3 to 9, and the wall thickness from 10 to 20 cm. As a result, 16,200 datasets were generated in this mode. The two-target mode is similar, except that instead of one object, there are two objects, introducing three additional parameters to the problem, including the target's two-dimensional location and permittivity. In this case, 58,320 datasets were generated. In Table 1, the parameters that were used for generating the dataset and their ranges are summarized. Also, allocated 70% of the dataset for training, 15% for the validation dataset, and 15% for the test dataset.

### 3. Numerical and Experimental Results

In this work, we used a deep neural network (DNN) to estimate both object and wall parameters concurrently. Python is used to implement the DNN algorithm and the

TensorFlow and Keras frameworks (<https://keras.io>). For this purpose, we presented a model in which the network input and backbone, which are the hidden layers of the neural network, are the same for single-target and two-target modes. However, the network output for these modes differs. In the single-target mode, we estimate five parameters: the target's two-dimensional location, wall permittivity, target permittivity, and wall thickness. As a result, we consider the number of neurons in the last layer to be five. In the case of two-targets, we estimate three additional parameters: the 2D location and the permittivity of the second target. We used the ReLU activation function for this neural network's first and middle layers but the linear activation function for the final layer. Equations (11) and (12) illustrate the activation of the ReLU and linear functions.

$$f(x) = \begin{cases} 0, & x \leq 0, \\ x, & x > 0, \end{cases} \quad (11)$$

$$f(x) = x. \quad (12)$$

We trained the network using a batch size of 20, and a learning rate of 0.001, as well as the Adam optimizer and the Mean Squared Logarithmic Error (MSLE) loss function, as defined as follows:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N (\log(y_i + 1) - \log(\hat{y}_i + 1))^2, \quad (13)$$

where  $y$  is the actual value,  $\hat{y}$  is the estimated value, and  $N$  is the total number of data. MSLE is the mean of the squared differences between the actual and estimated values after log transformation. We sequentially

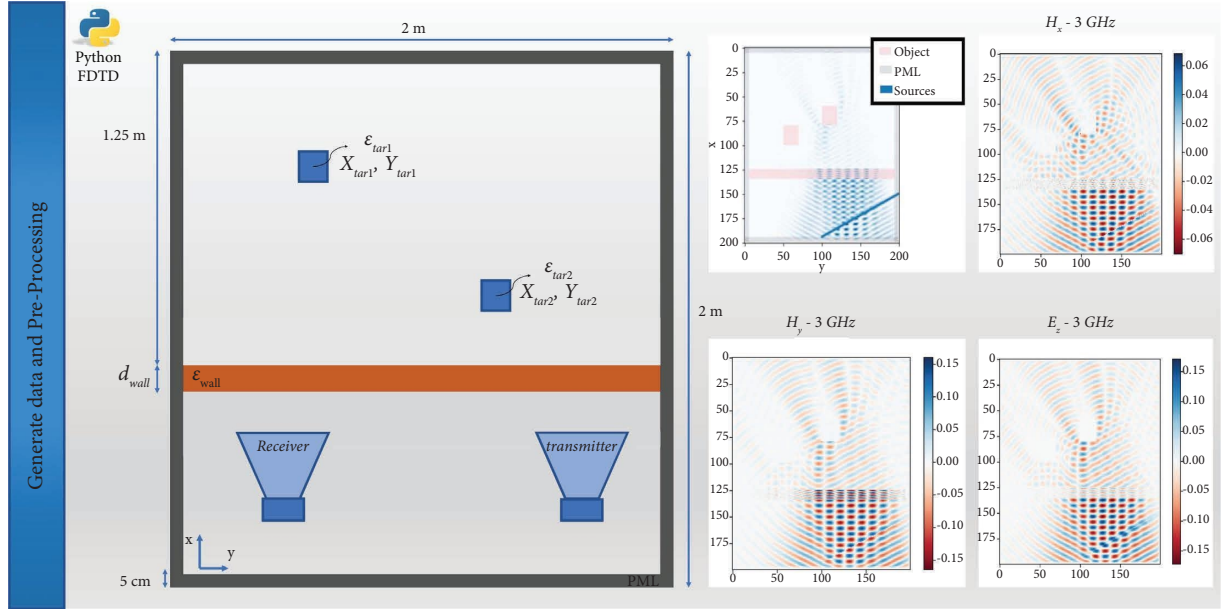


FIGURE 2: Environment and parameters required for generating the dataset. Simulations performed for the fields  $E_z, H_x, H_y$  are also available in the figure.

combined the Dense and Dropout layers to achieve higher accuracy and less loss in the network's backbone. The network is trained for 200 epochs, and the Loss diagram for the train and validation datasets are shown in Figure 3 in both single-target and two-target modes. Table 2 contains the accuracy and loss obtained on the validation and test datasets.

By including some target and wall specifications in the locating problem, we were able to improve the accuracy of the locating while also accurately estimating the wall and target material parameters. Indeed, each problem involves a large number of parameters. When we attempt to solve a problem using machine learning, if we enter all of the parameters involved in the problem, the model present by us can better learn the relationship between the inputs and outputs, thereby increasing the accuracy of the solution. We observed that it is sufficient to include additional parameters associated with the signal received by the receiver to achieve high positioning accuracy in the problem of two-dimensional positioning. By including the target and the wall's permittivity, and the wall's thickness in the problem, we discovered that the proposed model not only improved target location accuracy but also allowed for the integration of other critical parameters that had previously been estimated separately using a different algorithm. These parameters were estimated using the same deep learning model to locate the targets. The results of training network time, inference time, the size of deep learning models, and the number of deep neural network parameters are given in Table 3. The results are obtained using Google Colab and with a fixed GPU whose model is the Tesla K80 with 11 GB of RAM.

As described in this work, we tried to use the same backbone for a single or two targets. As shown in Figure 4, the input and backbone of the network are the same in

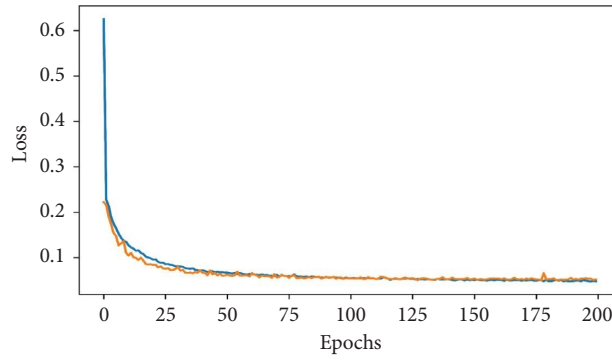
both cases. This theorem helps us to generalize the proposed algorithm to other modes using transfer learning. As discussed in the Transfer learning section, it can use the first and hidden layers in a specific scenario and train the model in another scenario with fewer data. Next, it can be applied the trained model for the single-target mode and then tried to use transfer learning by removing the last layer of the single-target model and replacing it with a new layer for the two-target mode. Here, all available data for the single-target mode is used first to train the model. The last layer was removed (five neurons) to estimate the parameters in Figure 4 for single-target mode and replaced with a layer with the eight neurons that were again used to estimate the parameters in Figure 4 for two-targets mode. Then we take 14000 data from 58320 data that we have for two-targets mode (about a quarter of the total data obtained through trial and error). After replacing the last layers of the model trained for the single-target, we train the model with this 14000 data for the two-targets mode.

The training results, in this case, are given in Figure 5 and Table 3. As you can see, with this technique, we could generalize the model trained for the single-target mode to the other mode with fewer data. We selected model parameters such as the learning rate of the loss function, etc., like the previously selected mode, and training has been done for 100 epochs.

The signal can also be added with noise to make it closer to reality. To measure the performance of the model in the presence of noise, we added an Additive White Gaussian Noise (AWGN) with different SNRs. Figure 6 shows the accuracy and loss value for the test dataset for single and two-target modes. A comparison of the results of this study with previous work is shown in Table 4.

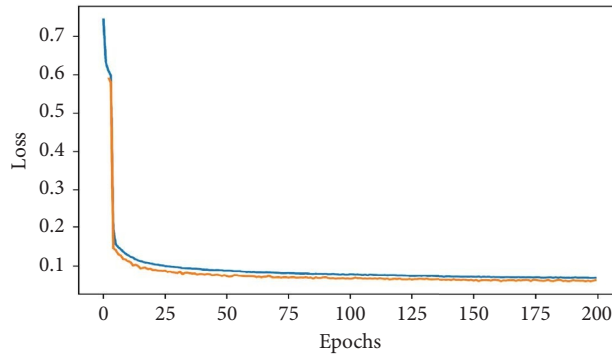
TABLE 1: A description of the dataset parameters and their values.

	Single-target			Two-target		
	Length	Step	N.O. Selected state	Length	Step	N.O. Selected state
Target permittivity 1	[5 to 85)	15	6	[5 to 85)	15	6
Target permittivity 2	—	—	—	[5 to 85)	15	6
Wall permittivity	[3 to 9)	1	6	[3 to 9)	1	6
Wall thickness (cm)	[10 to 20)	1	10	[10 to 19)	1	9
Target 1: $X_{\text{center1}}$	[5 to 85)	10	9	[5 to 85)	20	
Target 1: $Y_{\text{center1}}$	[50 to 100)	10	5	[50 to 100)	20	
Target 2: $X_{\text{center2}}$	—	—	—	[ $X_{\text{center1}} + 10$ to 90)	20	30
Target 2: $Y_{\text{center2}}$	—	—	—	[ $Y_{\text{center1}} + 10$ to 100)	20	
			Total datasets = 16,200			
						Total datasets = 58,320



— train loss  
— validation loss

(a)



— train loss  
— validation loss

(b)

FIGURE 3: Loss diagrams in single-target and two-target locating. (a) Single-target loss. (b) Two-target loss.

TABLE 2: Accuracy and loss results for different scenarios on evaluation and testing datasets.

	Single-target	Two-targets	Transfer learning mode
Accuracy on validation dataset	99%	98%	97%
Accuracy on test dataset	99%	99%	96%
Loss on validation dataset	0.052	0.065	0.081
Loss on test dataset	0.055	0.063	0.079

TABLE 3: Information of training time, inference time of one input, model size, and number of network parameters for both single and two targets.

	Single target	Two targets
Training time	8.1 min	29 min
Inference time	0.0017 sec	0.0018 sec
Model size	2.6 MB	2.7 MB
Number of network parameters	220115	220718

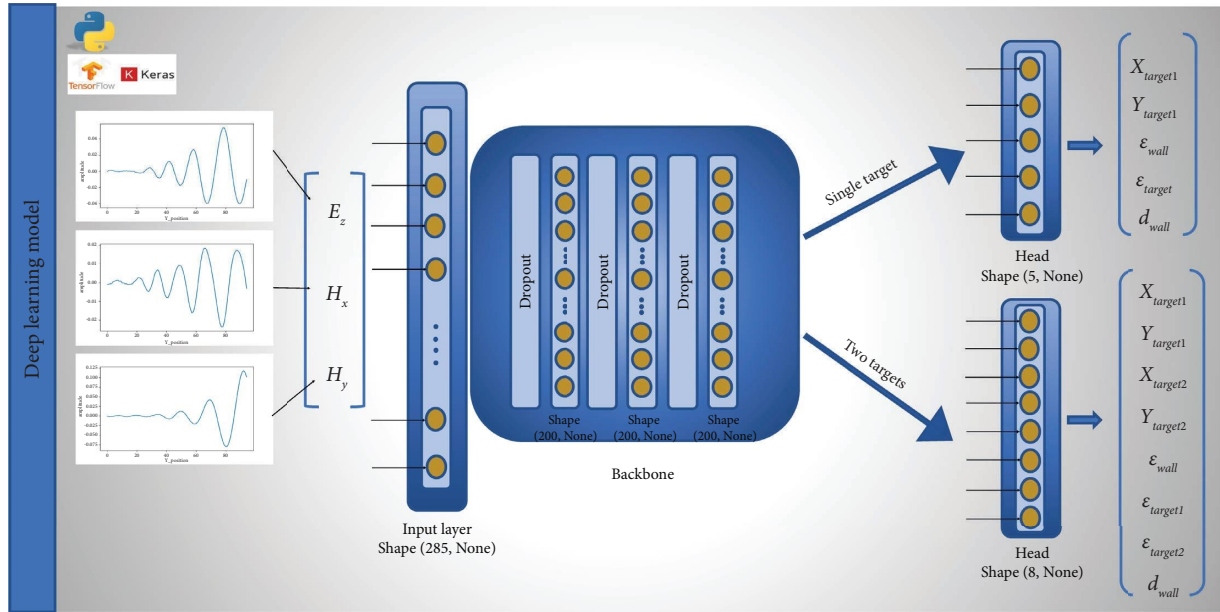
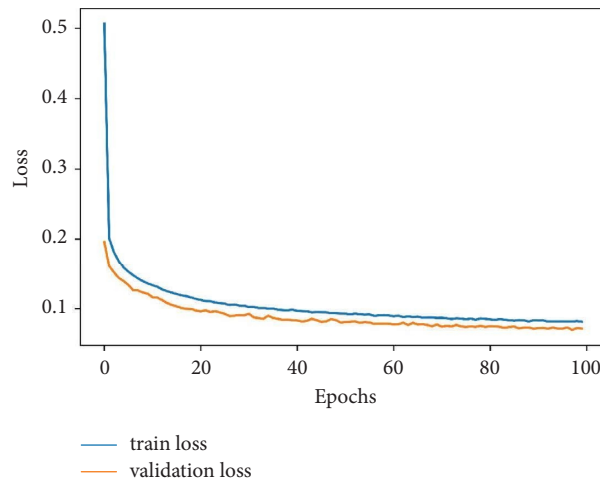
FIGURE 4: Overview of the deep neural network architecture presented. The network inputs are  $E_z$ ,  $H_x$ ,  $H_y$  fields and estimate the output according to one or two objectives or specified parameters for us.

FIGURE 5: Training and validation loss for two-target using transfer learning.



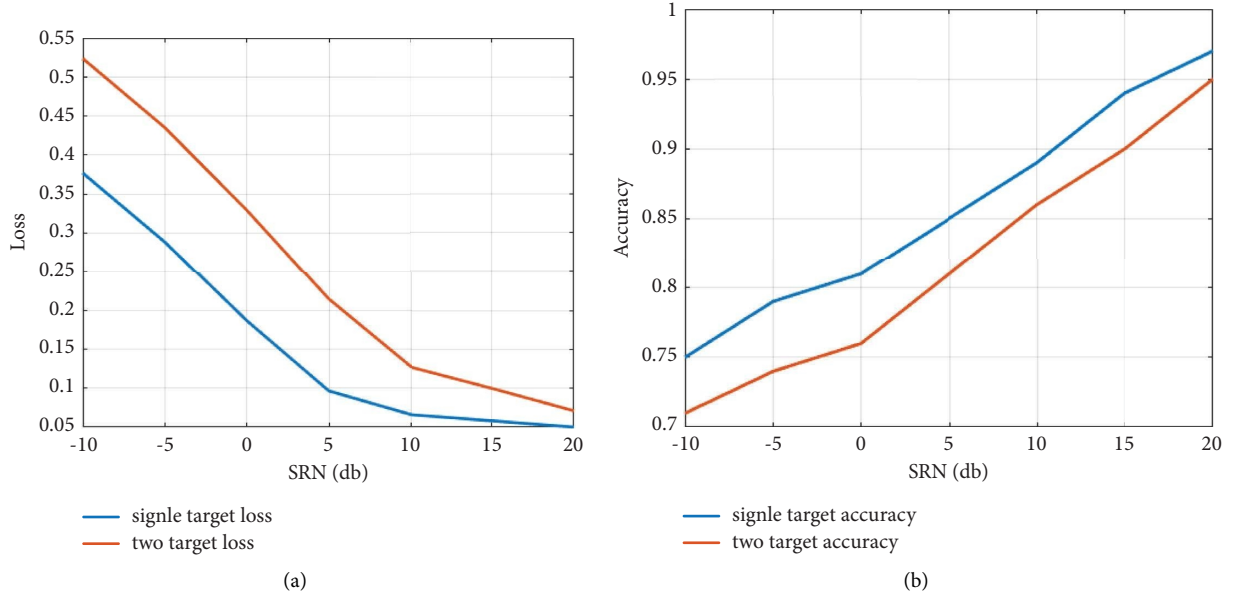


FIGURE 6: Comparison of accuracy and loss on test datasets for all single-target, two-target mode for different SNRs. (a) Loss diagrams. (b) Accuracy diagrams.

TABLE 4: A comparison of the results of this study with previous work.

Ref	Estimated parameters		Algorithm name	Metrics
	Wall	Object		
[20]	—	Wave number (k)	CNN	MAE = 6
[28]	$d_w, \epsilon_r$	—	SVM	$AE_{d_w} = 0.005$ $AE_{\epsilon_r} = 0.1$
[29]	$d_w, \epsilon_r, \sigma_w$	—	LS-SVM	$AE_{d_w} = 0.0002$
			SVM	$AE_{\epsilon_r} = 0.05$ $AE_{\sigma_w} = 0.0005$
[30]	—	$X_{dir}, Y_{dir}$	KEML	$RMSE_{X_{dir}} = 8.297$
			SVM	$RMSE_{Y_{dir}} = 2.22$
			LS-SVM	
This work	$d_w, \epsilon_r$	$X_{dir}, Y_{dir}, \epsilon_r$	Deep neural network	$MSLE_{SingleObject} = 0.052$ $MSLE_{TwoObjects} = 0.065$ $Accuracy_{SingleObject} = 99\%$ $Accuracy_{TwoObjects} = 98\%$

#### 4. Conclusions and Discussion

Estimating wall and object parameters has many benefits. In TWRI, it is challenging to obtain a clear image of the wall because of the ambiguity in its characteristics. Due to the proximity of the electromagnetic properties of the objects, such as their permittivity, there are always challenges in estimating objects and targets behind the wall. Since most furniture in a room has a permittivity of 5 to 15, but the human body has a permittivity that ranges from 80 to 90, it is usually difficult for radars to distinguish between these targets. Occasionally, they are separated by radar cross-section (RCS) [32]. The presented model addresses these challenges and will permit separating these targets with their permittivity and also simultaneously estimating wall parameters. Meanwhile, the presented deep learning model

is very small and can perform estimation in real-time with high speed, which makes it a good choice for TWRI applications.

This paper presents a model for simultaneously estimating the target and wall parameters using a deep neural network. Target parameters include the location and permittivity of the targets in two dimensions, as well as the thickness and permittivity of the wall. Two modes were considered, one with a single-target and one with two-targets, which required two parameters for the wall specification and three for each target. In this work, the dataset was generated by varying the parameters involved in the problem. Then, a deep learning model was presented that allows the parameters to be estimated for various targets by simply changing the model's end layer. By incorporating the parameters in the received signal into the receiver, the



locating accuracy was improved to 99% while simultaneously estimating parameters such as wall thickness and target and wall permittivity.

## Data Availability

The data that supports the findings of this study are available from the corresponding author upon reasonable request.

## Disclosure

An earlier version of this paper has been presented as a preprint according to the following link: <https://arxiv.org/abs/2111.04568>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

F.G. and H.S. conceived the idea. F.G. set up the DNN model. F.G. conducted the simulations for dataset. Finally, F.G. wrote the manuscript based on the input from all authors. H.S. supervised the project and reviewed the manuscript.

## Acknowledgments

## References

- [1] H. Li, G. Cui, S. Guo, L. Kong, and X. Yang, "Human target detection based on fcn for through-the-wall radar imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 9, pp. 1565–1569, 2021.
- [2] K. Wang, Z. Zeng, and J. Sun, "Through-wall detection of the moving paths and vital signs of human beings," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 717–721, 2019.
- [3] M. Fallahpour and R. Zoughi, "Fast 3-d qualitative method for through-wall imaging and structural health monitoring," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2463–2467, 2015.
- [4] M. Dehmollaian, M. Thiel, and K. Sarabandi, "Through-the-wall imaging using differential sar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1289–1296, 2009.
- [5] Y.-Z. Hu, T.-J. Li, and Z.-O. Zhou, "Use of the location inverse solution to reduce ghost images," *EURASIP Journal on Applied Signal Processing*, vol. 2010, pp. 454705–454707, 2010.
- [6] F. H. C. Tivive, A. Bouzerdoum, and M. G. Amin, "A subspace projection approach for wall clutter mitigation in through-the-wall radar imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2108–2122, 2015.
- [7] Y. Peng and S. Guo, "Detailed feature representation and analysis of low frequency uwb radar range profile for improving through-wall human activity recognition," in *Proceedings of the 2020 IEEE Radar Conference (RadarConf20)*, pp. 1–6, IEEE, Florence, Italy, September 2020.
- [8] L.-j. Kong, G.-l. Cui, J.-y. Yang, and X.-b. Yang, "Wall parameters estimation method for through-the-wall radar imaging," in *Proceedings of the 2008 International Conference on Radar*, pp. 297–301, IEEE, Adelaide, SA, Australia, September 2008.
- [9] S. Tripathi, S. R. Patre, S. Singh, S. Singh, and S. P. Singh, "Triple-band microstrip patch antenna with improved gain," in *Proceedings of the 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)*, pp. 106–110, IEEE, Sultanpur, India, March 2016.
- [10] A. Helen Victoria and G. Maragatham, "Activity recognition of fmcw radar human signatures using tower convolutional neural networks," *Wireless Networks*, pp. 1–17, 2021.
- [11] X. Chen and W. Chen, "Wall parameters estimation based on support vector regression for through wall radar sensing," *EURASIP Journal on Applied Signal Processing*, vol. 2015, no. 1, pp. 57–13, 2015.
- [12] F. Ghorbani, S. Hashemi, A. Abdolali, and M. Soleimani, "Eegsig machine learning-based toolbox for end-to-end eeg signal processing," 2020, <https://arxiv.org/abs/2010.12877>.
- [13] F. Ghorbani, S. Beyraghi, J. Shabanpour, H. Oraizi, H. Soleimani, and M. Soleimani, "Deep neural network-based automatic metasurface design with a wide frequency range," *Scientific Reports*, vol. 11, no. 1, pp. 7102–7108, 2021.
- [14] F. Ghorbani, J. Shabanpour, S. Beyraghi, H. Soleimani, H. Oraizi, and M. Soleimani, "A deep learning approach for inverse design of the metasurface for dual-polarized waves," *Applied Physics A*, vol. 127, no. 11, p. 869, 2021.
- [15] M. Soltani, F. Ghorbani, S. Beyraghi, Y. Azizi, and M. Soleimani, *Multifunctional Automatic Metasurface Design Using Deep Learning Approach*, TechRxiv, 2022.
- [16] Y. Sharma, H. H. Zhang, and H. Xin, "Machine learning techniques for optimizing design of double t-shaped monopole antenna," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 7, pp. 5658–5663, 2020.
- [17] R. Medico, N. Lambrecht, H. Pues et al., "Machine learning based error detection in transient susceptibility tests," *IEEE Transactions on Electromagnetic Compatibility*, vol. 61, no. 2, pp. 352–360, 2019.
- [18] K. K. Sharma, A. Gupta, B. Sharma, and S. L. Tripathi, *Intelligent Communication and Automation Systems*, CRC Press, Boca Raton, Florida, 2021.
- [19] K. Kumar, K. Chaudhury, and S. L. Tripathi, "Future of machine learning (ml) and deep learning (dl) in healthcare monitoring system," *Machine Learning Algorithms for Signal and Image Processing*, 2022.
- [20] A. Wood, R. Wood, and M. Charnley, "Through-the-wall radar detection using machine learning," *Results in Applied Mathematics*, vol. 7, Article ID 100106, 2020.
- [21] F. Ghorbani, H. Soleimani, and M. Soleimani, "Deep learning approach for target locating in through-the-wall radar under electromagnetic complex wall," 2021, <https://arxiv.org/abs/2102.07990>.
- [22] H.-M. Zhang, S. Zhou, C. Xu, and J. J. Zhang, "A robust approach for three-dimensional real-time target localization under ambiguous wall parameters," *Progress In Electromagnetics Research M*, vol. 97, pp. 145–156, 2020.
- [23] A. P. Singh, "A robust and efficient wall parameter estimation approach for through wall radar," *International Journal of Microwave and Wireless Technologies*, pp. 1–7, 2022.
- [24] R. Solimene, F. Soldovieri, G. Prisco, and R. Pierri, "Three-dimensional through-wall imaging under ambiguous wall parameters," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1310–1317, 2009.
- [25] P. Protiva, J. Mrkvica, and J. Machác, "Estimation of wall parameters from time-delay-only through-wall radar

- measurements," *IEEE Transactions on Antennas and Propagation*, vol. 59, no. 11, pp. 4268–4278, 2011.
- [26] T. Jin, B. Chen, and Z. Zhou, "Image-domain estimation of wall parameters for autofocusing of through-the-wall sar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, pp. 1836–1843, 2013.
  - [27] F. Fereidoony, S. A. Mirtaheri, and S. Chamaani, "M-sequence sensor and continuous basis estimator for wall parameter estimation utilizing through the wall sensing," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4083–4091, 2017.
  - [28] H.-M. Zhang, Y.-R. Zhang, F.-F. Wang, and J.-L. An, "Application of support vector machines for estimating wall parameters in through-wall radar imaging," *International Journal of Antennas and Propagation*, vol. 2015, pp. 1–8, 2015.
  - [29] H. M. Zhang, Y. R. Zhang, Z. B. Wang, Z. H. Wu, and C. X. Zhang, "An efficient method based on machine learning for estimation of the wall parameters in through-the-wall imaging," *International Journal of Remote Sensing*, vol. 37, no. 13, pp. 3061–3073, 2016.
  - [30] H.-M. Zhang, S. Zhou, C. Xu, and Y.-R. Zhang, "A real-time automatic method for target locating under unknown wall characteristics in through-wall imaging," *Progress In Electromagnetics Research M*, vol. 89, pp. 189–197, 2020.
  - [31] J. B. Schneider, "Understanding the finite-difference time-domain method," *School of electrical engineering and computer science*, Washington State University, vol. 181, , 2010.
  - [32] T. D. Bufler and R. M. Narayanan, "Radar classification of indoor targets using support vector machines," *IET Radar, Sonar & Navigation*, vol. 10, no. 8, pp. 1468–1476, 2016.