A deep-learning approach for through-the-wall radar tracking in 3D

By Gabriele Incorvaia and Oliver Dorn

Department of Mathematics, The University of Manchester, UK,
gabriele.incorvaia@manchester.ac.uk, oliver.dorn@manchester.ac.uk

Abstract

In this work, a data driven tracking scheme for through-the-wall applications is presented. We propose a strategy based on deep-learning aimed at localizing and following the trajectory of targets of interest hidden inside a building for surveillance applications. Firstly, we rewrite the tracking problem as a classification task. Then, starting from through-the-wall measurements, we tackle it by employing a combination of different Multi-Layer Perceptron networks. No a-priori dynamic assumptions are required, which distinguishes the proposed approach from popular Bayesian tracking filters. Numerical experiments are performed in 3D to assess the accuracy of the method, where the simulation of the propagating fields is modelled using an FDFD-approximation of Maxwell's equation in 3D.

1. Introduction

The object-tracking problem is generally defined as the mathematical task of identifying, localizing and characterizing the trajectory of moving objects over time Challa (2011). Nowadays, this task appears in a variety of applications, ranging from air traffic control to video sequence-based observations, including the localization of a person in a crowd Khan et al (2019), Nkwari et al (2018). In this work, we focus on through-the-wall radar tracking problems, namely identifying the trajectory of moving objects hidden behind walls. Our ultimate goal is the definition of a reliable tracking procedure for almost real-time localization of targets hidden inside a building for surveillance purposes.

The characterization of through-the-wall targets relies on indirect electromagnetic measurements performed by antennas located around the considered building.
Although Bayesian tracking approaches (e.g. Kalman filters, Particle Filters) Challa (2011) have been proven successful in these applications, we focus instead on a novel data-driven localization strategy based on deep-learning. A combination of neural networks is employed to retrieve information on the trajectories by independently localizing the objects of interest at consecutive time steps.
The idea of performing almost real-time tracking by combining distinct networks, trained on independent data, constitutes the main novelty of this work. Furthermore, no a-priori dynamic information is required in our approach, distinguishing it from more traditional Bayesian filters. Nevertheless, the possibility of adding a post-processing step in order to account for prior knowledge about the expected motion, when available, is also discussed.

The structure of this paper is as follows. We firstly introduce a mathematical model based on Maxwell's equations to approximate the propagation of the electromagnetic fields within the domain. Then, we describe the main features of the data driven tracking scheme presented here, with particular emphasis on the architecture of the network employed. Finally, some numerical experiments are performed in a 3D setup.

2. Maxwell's equations

The propagation of the electromagnetic fields is schematized by considering Maxwell's equations in the frequency domain as follows

$$\oint_B \mathbf{H} \cdot \mathbf{dl} = i\omega \int_S (\epsilon \mathbf{E}) \cdot \hat{\mathbf{n}} dS + \int_S \mathbf{J} \cdot \hat{\mathbf{n}} dS, \tag{2.1}$$

$$\oint_B \mathbf{E} \cdot \mathbf{dl} = -i\omega \int_S (\mu \mathbf{H}) \cdot \hat{\mathbf{n}} dS + \int_S \mathbf{M} \cdot \hat{\mathbf{n}} dS. \tag{2.2}$$

Here, $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields, ϵ and μ are the dielectric permittivity and the magnetic permeability (possibly diagonal dyads). S is an open surface with boundary B, $\mathbf{M}$ is the electric current density due to external sources and $\mathbf{J}$ is defined as the sum of the electric conductivity current, $\mathbf{J}_{cond} = \sigma \mathbf{E}$, and the electric current density due to electric external sources $\mathbf{J}_{ext}$. Here, σ is the electrical conductivity (possibly a diagonal dyad). Furthermore, we denote by $\hat{\mathbf{n}}$ the outward unitary normal to the surface considered.

In order to numerically implement the above model, the equations are discretized using a Finite-Differences-Frequency-Domain (FDFD) approximation using Python as programming language. Here, we follow closely the numerical scheme suggested in Champagne et al (2001). More details on our particular implementation will be presented elsewhere.

3. Through-the-wall radar target tracking

Characterizing and localizing targets of interest, starting from electromagnetic data, has been widely explored in the literature, see for example Poulton et al (1992). Traditional techniques such as the MUSIC algorithm, Cheney (2001), the linear sampling method, Colton et al (2003), Potthast et al (2006), and level-set schemes, Dorn et al (2000), Dorn et al (2006), have been proven to be able to retrieve objects embedded in known backgrounds in a variety of applications. More recently, various deep-learning-based strategies, Kujawski et al (2019), Niu et al (2019), Mostajabi et al (2019), have been successfully applied to similar problems.

Although traditional reconstruction strategies, usually incorporating a specific physical model as part of an optimization approach, are generally highly reliable and offer the possibility to incorporate prior physical information in form of a tailor-made regularization strategy. However, they are usually computationally expensive and therefore not suitable for real time tracking.

In this work, we focus on data-driven techniques which, after completing a training process, are computationally efficient and suited for online processing of incoming data. We highlight however that they also have some limitations. Among them, there is a strong dependence on the quality of the data set which is available for the training task. That should be rich enough to completely describe the phenomena under consideration, Adler et al (2017), a condition which may be difficult to satisfy for many applications where collecting data is expensive or time-consuming. Hybrid approaches, combining machine learning and inverse problems theory, have been proposed as well. For example, we refer to our previous work Incorvaia et al (2021) and references therein.

In this study, a proof-of-concept analysis is discussed for a machine-learning approach to this challenge. The core idea is to construct a network which maps the measurements (i.e. the electric field values at the receiver positions) directly to the target locations. Practically, this is realized by dividing the domain of interest into a number (N_c) of subregions by construction of a coarse grid and specifying in the tracking process in which cells of this grid the objects are located at each time step. Therefore, the localization problem is rewritten as a classification task, which is then addressed by using deep-learning strategies. More details are provided in the next section.

4. A machine-learning approach in 3D

The main concepts behind the development of the 3D data-driven tracking strategy introduced in this work are outlined in this section. The primary goal is to estimate the locations of moving objects hidden behind walls and following their movements over time. Data is (synthetically) collected in our proof-of-concept study by measuring the electric field values at receivers placed around the building of interest. However, due to the high computational cost of the forward model, a trade-off between a realistic setup and its corresponding computational simulation has been chosen here for ease of demonstration.

The procedure developed here relies on a combination of different types of classifiers. According to LeBlanc et al (1996), the main advantage of this combination is that multiple networks might be able to incorporate complementary information and boost the overall capability of the network. Furthermore, since the independence of the chosen classifiers plays a major role LeBlanc et al (1996), we choose to train them on independent data sets associated with distinct groups of sources. In more detail, each group includes antennas distributed around the walls of the building of interest, but at different locations. The corresponding fields are evaluated by a common set of receivers, which are as well located around the building. Eventually, a specific strategy is applied to combine the predictions made by each classifier, which will be explained further below.

Although we have anticipated that the input data is given by direct field measurements, a few clarifications are in order. Firstly, given a set of data, we pre-process this data set by subtracting from it the contributions given by the field values measured in empty building background (i.e. without any targets). This assumes that we do have access to that data in principle, e.g. data having been collected while there are no moving objects inside the building. Stationary objects are still permitted to be present during this phase, which will provide us with the stationary background during the tracking task. This data subtraction approach is not new and has been applied in many related applications with great success. Overall, it splits the tracking object from a stationary inverse problem formulation of estimating the general environment, and helps to improve the stability of the tracking method Incorvaia et al (2021). Once this step is completed, the mathematical norms of the resulting complex data values are computed and fed as real numbers into the networks.

More technical details on the network structure are provided next. According to Goodfellow et al (2016), the selection of the architecture of a network is generally problem-specific and the key point is to match its complexity with the complexity of the data Puzyrev (2019). In addition, this choice should be motivated by the type of data considered and by the nature of the features involved. Overall, this is expected to introduce a

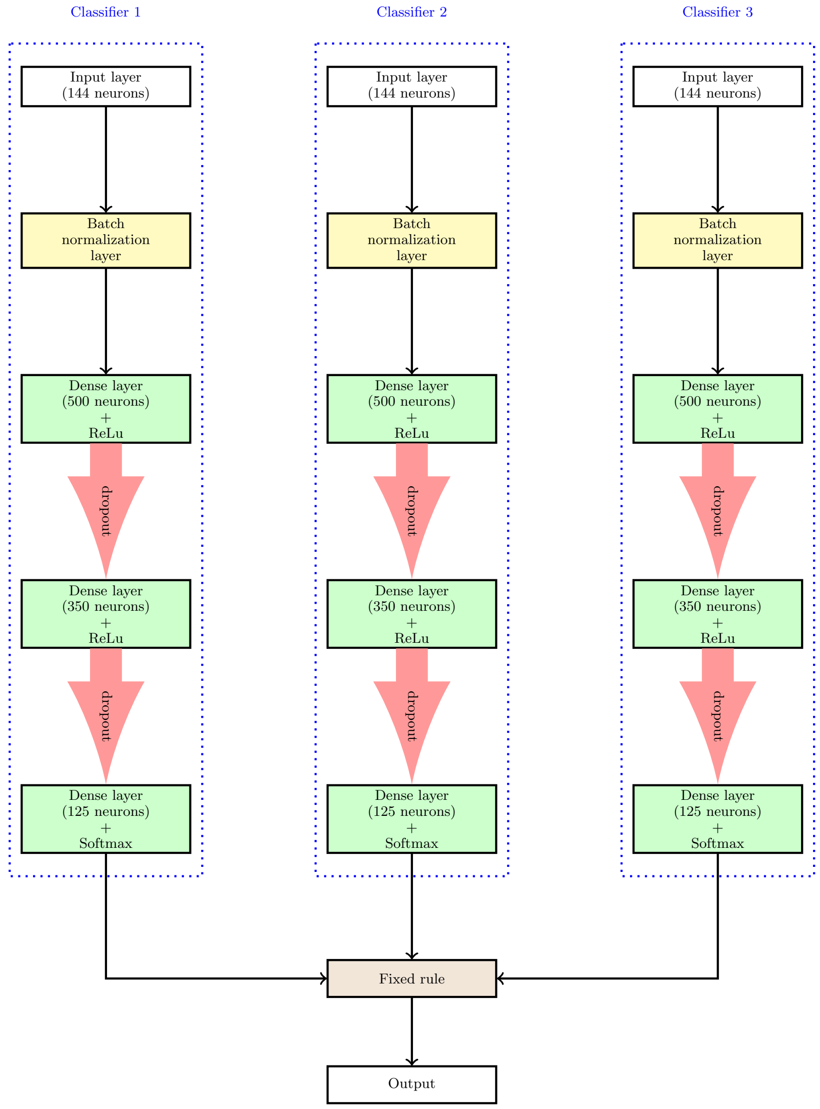
Classifier 1
Classifier 2
Classifier 3
Input layer (144 neurons)
Batch normalization layer
Dense layer (500 neurons) + ReLu
dropout
Dense layer (350 neurons) + ReLu
dropout
Dense layer (125 neurons) + Softmax
Fixed rule
Output

Figure 1. Network architecture. The blocks corresponding to each MLP classifier are highlighted. Their outcomes are combined by using a pre-defined fixed rule as described in the text.

positive regularization effect which helps to increase the global performance Kukacka et al (2017).

5. Specific Details of the Network

In this work, three independent but identical Multi-Layer-Perceptron (MLP) networks are considered and eventually combined by applying a fixed rule strategy inspired by probabilistic arguments. See Figure 1.

Following general ideas provided in Incorvaia et al (2021), a brief mathematical characterization of these MLPs is provided next. Let L be the total number of layers and n_l be the number of neurons included in the l-th layer, $l = 1, ..., L$. Following Higham et al (2019), we denote by $W^{[l]} \in \mathbf{R}^{n_l \times n_{l-1}}$ and $b^{[l]} \in \mathbf{R}^{n_l}$ the matrix of weights and the vector of biases of the l-th layer, respectively, given an input $x \in \mathbf{R}^{n_1}$. This network can be seen as a map from $\mathbf{R}^{n_1}$ to $\mathbf{R}^{n_L}$ such that

$$a^{[1]} = x,$$

$$a^{[l]} = \sigma(W^{[l]} a^{[l-1]} + b^{[l]}), \quad l = 2, ..., L. \tag{5.1}$$

Here, $a^{[l]} \in \mathbf{R}^{n_l}$ denotes the activation (i.e. outcome) of the l-th layer while $\sigma(\cdot)$ represents an element-wise non-linear activation function.

Following Higham et al (2019), given a training set $\{x^{(i)}\}_{i=1}^{N}$, let us assume that each data point $x^{(i)}$ is associated to a label $l_i \in \{1, ..., K\}$ specifying the class of the sample considered out of the K possible choices. Moreover, let $v^{(i)} \in \mathbf{R}^K$ be the outcome of a generic network for the input data $x^{(i)}$, assuming that its j-th component is large when $x^{(i)}$ is likely to belong to the j-th class. Then the application of the softmax activation function $\sigma_{SM}(\cdot)$ on $v^{(i)}$ yields

$$\sigma_{SM}(v_j^{(i)}) = \frac{\exp v_j^{(i)}}{\sum_{m=1}^{K} \exp v_m^{(i)}}, \quad j = 1, ..., K. \tag{5.2}$$

Each outcome $\sigma_{SM}(v_j^{(i)})$ can be seen as the probability of the data point $x^{(i)}$ to belong to the j-th class, estimated by the network.

In order to reliably identify the true class of each example considered, a categorical cross-entropy loss function is chosen

$$loss = -\sum_{i=1}^{N} \log \left[\frac{\exp v_{l_i}^{(i)}}{\sum_{j=1}^{K} \exp v_j^{(i)}} \right]. \tag{5.3}$$

As shown in Figure 1, a softmax activation function is selected in the outcome layer of each MLP. However, a ReLU activation function

$$\sigma_{ReLU}(x) = \max\{x, 0\}, \quad x \in \mathbf{R}, \tag{5.4}$$

is used for their hidden layers.

In addition, batch normalization operations are realized to improve stability, and dropout layers Goodfellow et al (2016) are introduced to counter over-fitting. As discussed in Srivastava et al (2014), these layers promote neuron independence since the final network parameters can be considered as the result of an average of learning processes realized on networks with (slightly) different architectures. Overall, the structure of this network is a generalization to this new 3D situation of the one presented in our previous paper Incorvaia et al (2021), where it has been developed and implemented for a 2D situation.

Each classifier is trained individually and independently. A supervised learning approach is considered in which the trainable parameters of the network are assigned according to the so-called Adam optimization algorithm Kingma et al (2014), aimed at reducing the value of the loss function Puzyrev (2019).

Our description of the structure of the network is completed by specifying the procedure followed here to combine the estimations given by each classifier. The combination of multiple simple networks has been successfully employed in several applications Ji et al (1997), Alexandre et al (2001). It has been shown capable to often outperform the use of a large single classifiers in terms of both accuracy and stability Sharkey et al (1995). This approach might also be computationally advantageous for other reasons, including that independent MLPs might be trained in parallel and that it avoids the expensive task of training one single very large network.

According to LeBlanc et al (1996), Mohandes et al (2018), the combination of individual classifiers also tends to improves the generalization capabilities since it may merge complementary information. This usually happens if the networks considered are trained on different features or on independent data sets, as it is the case here. In our application, the combination is based on the so-called product- and sum-rules Tax et al (1997). For each cell, they essentially consist of multiplying or summing the confidence scores (i.e. an approximation of the probability of the cell to contain the target) given by each classifier and eventually selecting the cell with the highest combined value.

To be more precise, let $y_1, y_2, y_3 \in \mathbf{R}^{N_c}$ be the (softmax) outcomes of each MLP. Then, the combined prediction $\tilde{y} \in \mathbf{R}^{N_c}$ is computed as follows

$$\tilde{y}^{(i)} = \frac{y_1^{(i)} y_2^{(i)} y_3^{(i)}}{N_p}, \text{ with the product rule,}$$

$$\tilde{y}^{(i)} = \frac{y_1^{(i)} + y_2^{(i)} + y_3^{(i)}}{N_s}, \text{ with the sum rule,} \tag{5.5}$$

where $i = 1, ..., N_c$ and N_p, N_s are normalization factors. Hence, the location estimated for the target is the k-th cell, where

$$\tilde{y}^{(k)} = \max_{j \in [1, N_c]} \tilde{y}^{(j)}. \tag{5.6}$$

6. Numerical experiments

The accuracy of the 3D tracking scheme introduced in this paper is evaluated by performing numerical experiments as discussed in the following.

We start by describing the setup considered here. Our choice is a trade-off between the goal to consider a realistic scenario and the necessity of limiting the computational cost for such a proof-of-concept study. We schematize a domain (building) with dimensions 21.5m × 21.5m × 21.5m using a 3D-grid containing 43 × 43 × 43 cubic voxels. Each of those voxels has edge length $l = 0.5$m in all directions. This way, we model a building with size 7.5m × 7.5m × 7.5m by specifying its external walls. Each wall has a thickness of 0.5m and an electrical conductivity $\sigma_{wall} = 0.03$S/m. A single moving target with a conductivity value $\sigma_{target} = 1.0$S/m is considered inside this domain for generating the training and test data. The other electromagnetic parameters are fixed to the following background values: $\mu = \mu_0 = 4\pi \times 10^{-7}$H/m, $\epsilon = \epsilon_0 = 8.85 \times 10^{-12}$F/m, $\sigma_{bg} = 0.01$S/m.

All antennas are located just outside the lateral sides of the building but not on top or underneath it. The sources are overall divided into three groups, each of which includes 18 elements. A total of 144 receivers is considered. The antennas operate at a single frequency, which is chosen here (somehow arbitrarily) as $f = 100$MHz. A coarse grid is introduced to divide the domain between the outer building walls into $N_c = 125$ cubic cells, each containing $3 \times 3 \times 3$ voxels. This will be used to specify the position of the target by identifying the cell in which it is contained, according to our network estimation.

Due to the lack of real data (e.g. experimental measurements), the training set here used is synthetically generated by numerically solving the Maxwell model introduced before. In more details, assuming the presence of a single object inside the domain at a known position, the associated electric field values at the receivers are approximated by solving a discretized version of Equations (2.1) and (2.2) numerically. Furthermore, in

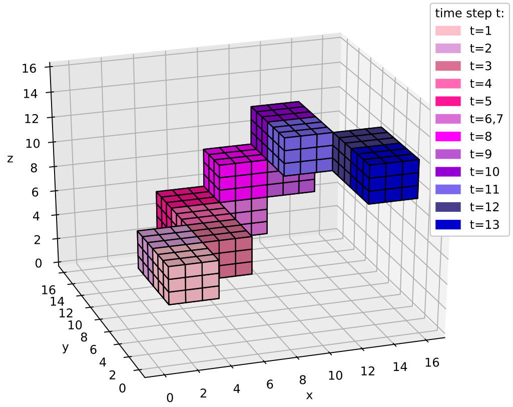

FIGURE 2. 3D trajectory described by a moving target inside a building. For each time step, the cell of the coarse grid containing the target is highlighted. The walls of the building are not shown for clarity.

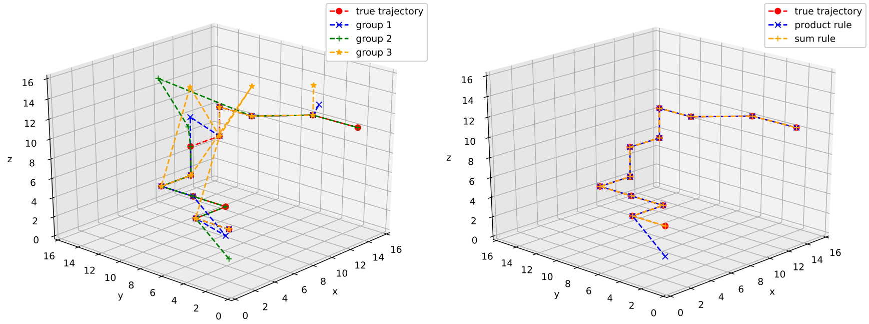

FIGURE 3. To the left: estimations attained by considering each MLP singularly. To the right: estimations attained by combining the classifiers. For each time step, the center of the predicted cell is highlighted and compared to the true trajectory reported in red.

order to increase the variability of the training data set, different object sizes and shapes are considered during this process.

The richness of the data set is further increased by adopting a data augmentation technique which, according to Goodfellow et al (2016), might help to prevent over-fitting issues. It consists of synthetically expanding the (training) data set by applying transformations to the original samples DeVries et al (2017). In this work, the generation of new samples is realized by adding small-noise perturbations to the original ones.

From a computational viewpoint we mention that the simulations related to the network training are run on a Nvidia V100 GPU, where the network implementation is realized using the Keras library available for Python. The simulations related to solving (2.1) and (2.2) numerically are done on a standard parallelized CPU architecture.

The first numerical example considered here aims at following the motion of a single target whose trajectory is illustrated in Figure 2. The corresponding outcomes of the proposed tracking procedure are shown in Figure 3.

Although the predictions of each single classifier provide reasonable indications of the followed trajectory, a few errors and misinterpretations occur. However, as expected, different networks make mistakes of different nature. Hence, their combination is hoped to

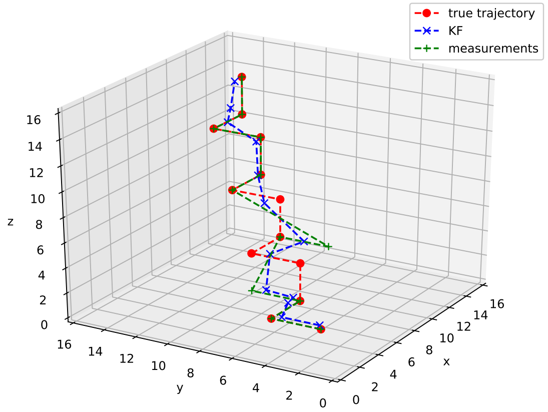

FIGURE 4. A post-processing step based on a Kalman filter is considered to account for prior dynamic knowledge. The outcomes of the MLP network, the results of the Kalman filter and the true trajectory are illustrated in green, blue and red respectively.

improve the overall accuracy of the method. This is confirmed by the right plot in Figure 3 which shows the combination of the individual estimates according to our scheme.

The effectiveness of the proposed scheme is clearly visible from the previous example. However, a natural question arises whether the improvement of accuracy reached is actually due to the combination strategy or simply a consequence of the increased number of antennas considered. To shed some light on this matter, we have used the superimposition principle to sum all measurements associated with different source groups and train a single MLP network on this new enhanced data set. This offers a performance which is comparable with the ones of the single MLPs when trained on each group of measures. This confirms, as expected, that it is not the simple increase of the number of sources considered here, but instead the combination of independent types of information which improves the overall estimation.

We highlight again that the previous results have been obtained without requiring any a priori dynamic information. This represents an advantage of our technique compared to standard Bayesian filters in situation where such dynamic information is not available or where it might be misleading. However, we recognize that there may be cases where this (prior) information is available. Hence, it would be beneficial to account for it to boost even further the tracking performance.

We show in the following that this is possible also with our proposed network by using, for example, a post-processing step based on the application of a Kalman filter Challa (2011), similar to a tracking scheme introduced in a different context in Incorvaia et al (2020). We briefly recall that the Kalman filter is a popular tracking tool which operates in two steps. Firstly, in the prediction step a forecast is performed predicting the next target location according to a dynamic model. Secondly, based on that forecast and on any available new data, in an update step this forecast is corrected by considering the new data at the current time.

As for the dynamic model, we adopt here a Continuous White Noise Acceleration (CWNA) model Challa (2011) (assuming an almost constant velocity) describing an object moving inside the building along its main diagonal. In our machine-learning approach, the incoming measurements are given by the outcomes of a MLP classifier trained on the data associated with a single group of sources (i.e. 18 sources in total), as before. The corresponding results are illustrated in Figure 4. This simple example shows that the Kalman filter post-processing step helps to further counter the effects of wrong MLP estimations in a situation where additional dynamic information is available. Figure 4 makes us confident that a similar approach might reliably work even in more realistic situations where, for example, multiple targets are present or more complicated motions occur. We plan to further investigate such more complex scenarios in our future research.

7. Conclusions

In this work, we have presented a data-driven strategy developed for localizing and tracking targets of interest hidden behind walls. Our approach is based on a numerical FDFD implementation of Maxwell's equations to model the propagation of the fields within the domain of interest. Then, we have presented a novel data-driven tracking approach aimed at mapping through-the-wall radar measurements directly to target locations and thereby retrieving the target trajectory in almost real time. A combination of Multi-Layer Perceptron networks has been employed, of which the overall architecture has been described. By performing numerical experiments in 3D, the accuracy of the proposed method has been demonstrated in a simple simulated but sufficiently realistic test case. Finally, a post-processing technique has been proposed based on a Kalman Filter technique for situations where some general knowledge on the dynamics of the target is available. Overall, the results obtained in this proof-of-concept study make us confident that such a procedure can be successful even in real-world scenarios where potentially more complicated situations need to be addressed.

ACKNOWLEDGMENT

The work of GI was supported by the Defence Science and Technology Laboratory (Dstl) under the PhD sponsorship contract number DSTLX-1000119341-4.

REFERENCES

Subhash Challa et al 2011 Fundamentals of object tracking *Cambridge University Press.*

Gulraiz Khan, Zeeshan Tariq, and Muhammad Usman Ghani Khan 2019 Multi-Person Tracking Based on Faster R-CNN and Deep Appearance Features *Visual Object Tracking in the Deep Neural Networks Era. IntechOpen .*

PKM Nkwari, S Sinha, and HC Ferreira 2018 Through-the-wall radar imaging: a review *IETE Technical Review 35.6.*

Nathan J Champagne II, James G Berryman, and H Michael Buettner 2001 FDFD: A 3D finite-difference frequency-domain code for electromagnetic induction tomography *Journal of Computational Physics 170.2.*

Thomas Weiland 1984 On the unique numerical solution of Maxwellian eigenvalue problems in three dimensions *Part. Accel. 17.DESY-84-111.*

Henk A Van der Vorst 1992 Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems *SIAM Journal on scientific and Statistical Computing 13.2.*

Richard Barrett et al 1994 Templates for the solution of linear systems: building blocks for iterative methods. *Vol. 43. Siam.*

Mary M Poulton, Ben K Sternberg, and Charles E Glass 1992 Location of subsurface targets in geophysical data using neural networks *Geophysics 57.12.*

Margaret Cheney 2001 The linear sampling method and the MUSIC algorithm *Inverse problems 17.4.*

David Colton, Houssem Haddar, and Michele Piana 2003 The linear sampling method in inverse electromagnetic scattering theory *Inverse problems 19.6.*

Roland Potthast 2006 A survey on sampling and probe methods for inverse problems *Inverse Problems 22.2.*

Oliver Dorn, Eric L Miller, and Carey M Rappaport 2000 A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets *Inverse problems 16.5.*

Oliver Dorn and Dominique Lesselier 2006 Level set methods for inverse scattering *Inverse Problems 22.4.*

Adam Kujawski, Gert Herold, and Ennes Sarradj 2019 A deep learning method for grid-free localization and quantification of sound sources *The Journal of the Acoustical Society of America 146.3.*

Haiqiang Niu et al 2019 Deep-learning source localization using multi-frequency magnitude-only data *The Journal of the Acoustical Society of America 146.1.*

Amirhossein Mostajabi et al 2019 Single-sensor source localization using electromagnetic time reversal and deep transfer learning: application to lightning *Scientific reports 9.1.*

Jonas Adler and Ozan Oktem 2017 Solving ill-posed inverse problems using iterative deep neural networks *Inverse Problems 33.12.*

Gabriele Incorvaia and Oliver Dorn 2021 A deep-learning classifier for object tracking from through-the-wall radar data *2021 15th European Conference on Antennas and Propagation (EuCAP), to appear. IEEE.*

Michael LeBlanc and Robert Tibshirani 1996 Combining estimates in regression and classification *Journal of the American Statistical Association 91.436.*

Ian Goodfellow, Yoshua Bengio, and Aaron Courville 2016 Deep learning *MIT press.*

Vladimir Puzyrev 2019 Deep learning electromagnetic inversion with convolutional neural networks *Geophysical Journal International 218.2.*

Jan Kukacka, Vladimir Golkov, and Daniel Cremers 2017 Regularization for deep learning: A taxonomy *arXiv preprint arXiv:1710.10686.*

Catherine F Higham and Desmond J Higham 2019 Deep learning: An introduction for applied mathematicians *SIAM Review 61.4.*

Nitish Srivastava et al 2014 Dropout: a simple way to prevent neural networks from overfitting *The journal of machine learning research 15.1.*

Diederik P Kingma and Jimmy Ba 2014 Adam: A method for stochastic optimization *arXiv preprint arXiv:1412.6980.*

Chuanyi Ji and Sheng Ma 1997 Combinations of weak classifiers *Advances in Neural Information Processing Systems.*

Luìs A Alexandre, Aurèlio C Campilho, and Mohamed Kamel 2001 On combining classifiers using sum and product rules *Pattern Recognition Letters 22.12.*

Amanda JC Sharkey and Noel E Sharkey 1995 How to improve the reliability of artificial neural networks *Citeseer.*

Mohamed Mohandes, Mohamed Deriche, and Salihu O Aliyu 2018 Classifiers combination techniques: A comprehensive review *IEEE Access 6.*

David MJ Tax, Robert PW Duin, and Martijn Van Breukelen 1997 Comparison between product and mean classifier combination rules *Proc. Workshop on Statistical Pattern Recognition.*

Terrance DeVries and Graham W Taylor 2017 Dataset augmentation in feature space *arXiv preprint arXiv:1702.05538.*

Gabriele Incorvaia and Oliver Dorn 2020 Tracking targets from indirect through-the-wall radar observations *2020 14th European Conference on Antennas and Propagation (EuCAP). IEEE.*