Landmine Detection from GPR Data Using Convolutional Neural Networks

Silvia Lameri[1], Federico Lombardi[2], Paolo Bestagini[1], Maurizio Lualdi[3], Stefano Tubaro[1]

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano - Milano, Italy
[2]Department of Electrical and Electronic Engineering, University College London - London, United Kingdom
[3]Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano - Milano, Italy

Abstract—The presence of buried landmines is a serious threat in many areas around the World. Despite various techniques have been proposed in the literature to detect and recognize buried objects, automatic and easy to use systems providing accurate performance are still under research. Given the incredible results achieved by deep learning in many detection tasks, in this paper we propose a pipeline for buried landmine detection based on convolutional neural networks (CNNs) applied to ground-penetrating radar (GPR) images. The proposed algorithm is capable of recognizing whether a B-scan profile obtained from GPR acquisitions contains traces of buried mines. Validation of the presented system is carried out on real GPR acquisitions, albeit system training can be performed simply relying on synthetically generated data. Results show that it is possible to reach 95% of detection accuracy without training in real acquisition of landmine profiles.

I. INTRODUCTION

Landmines and explosive remnants of war contaminate large areas in more than 90 countries across the World, representing a serious and ongoing threat to civilians [1]. The figures of casualties due to landmines are not precisely known, but it is estimated that approximately 26.000 people a year are killed or maimed by landmines. Therefore, the development of methodologies to localize landmines for clearance of landmine sites is of paramount importance.

A possible way of solving landmine localization problem is to proceed in two separate steps: (i) buried object detection and (ii) object classification. Specifically, object detection consists in individuating the presence of buried targets that represent a possible threat. Object classification is the process of discriminating objects of interest (e.g., landmines) from other buried targets (e.g., clutter) [2]. In this paper, we focus on the first step.

As a matter of fact, landmine detection is a challenging problem since it is compounded by different factors: the large variety of landmine types, different soils conditions, weather conditions, presence of human and natural waste to name a few. Traditional fielded approaches use electromagnetic induction (EMI) based sensors specifically designed to detect metal targets. However, many modern landmines are made of plastic and contain little or no metal. In this context, ground

penetrating radar (GPR) systems have emerged as a suitable sensing modality for finding plastic threats [3], [4], [5]. Indeed, GPR sensors operate by measuring the reflection of an electromagnetic pulse from discontinuities in subsurface dielectric properties, thus they are able to detect nonmetal targets from their dielectric contrast with the soil environment. However, the sensitivity of GPR sensors to changing subsurface has also some drawbacks. Indeed, GPRs tend to detect the presence of clutter and soil distortion.

In this work we address the task of buried object detection in GPR data. In the literature many GPR signal and image processing techniques have been proposed for the automated detection of buried objects. Generally, these methods first implement a data pre-processing step that performs task as data normalization, correction for variations in depth and speed, removal of stationary effects due to the system response, background subtraction [6], [7], [8]. Then processed data is analyzed to detect the presence of buried targets. To this purpose, both model-based detection methods and features-based techniques have been proposed. Typical model-based approaches aim to individuate hyperbola in GPR images by making use of Hough transform [9] or fitting techniques [10]. However, the sensitivity of GPR systems to changes in local environmental conditions results in highly variable responses from buried objects that hinders correct hyperbola detection. In this scenario, detection algorithms based on statistical feature extraction from GPR images, including edge histogram descriptors [11], histograms of oriented gradients [12], hidden Markov models [13] among others, proved to have robust performance to a wide variety of data.

In this paper we propose an algorithm for landmine detection exploiting convolutional neural networks (CNNs) [14] for the analysis of GPR B-scans (i.e., 2D images of vertical underground slices). Our approach belongs to feature-based techniques category, but reverses the typical used paradigm. Indeed, we make use of a data-driven methodology that learns features characterizing buried targets directly from GPR images, rather than imposing any model or hand-crafted feature recipe, as done in [15] for landmine identification. In particular, we focus on a pipeline that necessitates of minimal image pre-processing (i.e., only track synchronization and removal of the direct antennas path) testing also different CNN architectures. The main advantages of using the proposed

approach with respect to other state-of-the-art solutions are highlighted by our experimental campaign carried out on GPR real data acquired from a test site. More specifically: (i) as our algorithm does not rely on any analytical modeling, it is less prone to errors due to simplistic assumptions or model simplifications (e.g., linearizations, etc.); (ii) the proposed method is able to work on small image patches with high accuracy, paving the way to precise target localization; (iii) the proposed CNN trained only on synthetic target signatures learns a feature extraction methodology that generalizes well on real GPR data; (iv) the possibility of embedding also real acquisitions in the training step enables to improve system performance up to 95% of accuracy.

II. Background

In this section we provide a sufficient background on GPR data acquisition and CNNs useful to understand the rest of the paper.

GPR Data Acquisition. For a given spatial position, a GPR transmit antenna emits an electromagnetic pulse into the ground and a receiving antenna measures the return signal's amplitude as a function of time. This single waveform recorded by the GPR with antennas at a given fixed position is referred to as A-scan. The structure of an A-scan is strongly affected by the medium through which the radiation propagates. If the medium contains regions with different dielectric constants, the A-scan will exhibit complex reflections at the region interfaces. When moving the GPR antennas on a line, one can gather a set of A-scans, which form a two dimensional data set, called a B-scan. In short, a B-scan is an image representing a vertical slice of the ground in which pixel intensities represent the amplitude of the received signal. Therefore, B-scans provide a more effective means for visualizing and characterizing a subsurface environments. Typically, two different patterns can be observed in a B-scan: (i) hyperbola signatures that derive from the reflection of the electromagnetic signal on small buried targets; (ii) linear segments due to the change of impedance between soil layers.

Convolutional Neural Networks. Convolutional neural networks (CNNs) are complex computational models that consist of a very high number of interconnected nodes associated to numeric parameters that can be tuned to learn complex and non-linear functions [14], [16]. Network nodes are stacked into multiple layers, each one performing a simple operation on its input. CNN layers typically comprise:

- *Convolution*: each convolution layer is a bank of filters h. Given an input signal x, the output of each filter is the valid part of the linear convolution.
- *Max pooling*: this layer downsamples the input x by sliding a small window over it and keeping the maximum value for each window position.
- *ReLU*: Rectified Linear Unit (ReLU) applies the rectification function $\max(0, x)$ to the input x, thus truncating negative values to zero [17].

- *Inner Product*: performs a set of linear combinations of all samples of the input x.
- *SoftMax*: normalizes the input values in the range $[0, 1]$ and guarantees that they sum up to one. This is particularly useful at the end of the network in order to interpret its outputs as probability values.

By feeding the CNN with a set of labeled data (e.g., images belonging to different known categories) and minimizing a cost function at the output of the last layer, CNN weights (e.g., the values of the filters in the convolutional layers, etc.) are tuned so that the CNN learns how to automatically extract distinctive features from data (e.g., image categories).

In image classification scenarios, the first networks layers usually learn low-level visual concepts such as edges and simple shapes, whereas deeper layers identify complex visual patterns. Finally, the last layer consists of a set of data that are combined using a given cost function that needs to be minimized. For example, in the context of binary image classification, the last layer is composed by 2 nodes (i.e., one per class), which define a probability distribution over the visual categories. That is, the value of a given node belonging to the last layer represents the probability of the input image to belong to that visual class.

To train a CNN model for a specific image classification task we need: (i) to define the metaparameters of the CNN, i.e., the sequence of operations to be performed, the number of layers, the number and shape of the filters in convolutional layers, etc; (ii) to define a proper cost function to be minimized during the training process; (iii) to prepare a (possibly large) dataset of training and test images, annotated with labels according to the specific tasks (i.e., GPR B-scans in our work).

III. Detection System

The goal of the proposed system is to detect whether a B-scan obtained through GPR acquisitions contains traces of buried objects for landmine detection. Formally, this means taking as input an image I representing a B-scan, and output a label $\hat{l}$ indicating possible absence (i.e., $\hat{l} = 0$) or presence (i.e., $\hat{l} = 1$) of objects.

The rationale behind the proposed technique is that B-scans present characteristic hyperbolic traces when GPRs analyze profiles over buried objects, as shown in Section II. Conversely, if the ground is relatively objects free, B-scans do not show prominent hyperbolas. It is therefore possible to leverage an image recognition system based on CNNs to discriminate between B-scans containing these traces or not.

The pipeline of the proposed detection system is sketched in Fig. 1. First, a CNN is trained to discriminate image patches containing object traces (i.e., hyperbolas) or not (i.e., background). When the system is trained, in order to detect whether an object is buried, a B-scan is acquired and split into patches. Each patch is tested against the CNN model. Votes associated by the CNN to each patch are aggregated into the final result. In the following, we present a detailed description of each step.
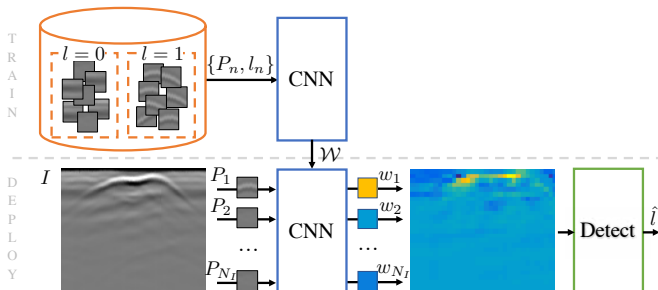
Fig. 1: Detection system pipeline. Training process on top, system deployment on bottom.

System Training Given a CNN architecture $\mathcal{N}$ we need to determine its set of weights $\mathcal{W}$ (i.e., filters coefficients, inner product weights, etc.) for the specific task. This is done training the CNN as in a standard supervised two-class problem. We make use of a database of $B \times B$ size patches P_n, $n \in [1, N_{\text{train}}]$ divided into two categories, i.e., *object* vs. *background* (see Fig. 1). Each patch is associated with a label l_n depending on its category: *background* patches P_n containing only background noise are labeled with $l_n = 0$; *object* patches P_n containing portions of hyperbola are labeled with $l_n = 1$. The CNN is fed with all available pairs $\{P_n, l_n\}$, $n \in [1, N_{\text{train}}]$ and learns to associate labels to patches.

Once the CNN is trained, it can be used to classify (i.e., associate to a label) new patches never used in the training step. Specifically, it is possible to feed the CNN with an unlabeled patch P_n, and obtain a vote w_n proportional to the likelihood of P_n to belong to class *object*. The higher w_n, the more likely P_n contains a portion of hyperbola.

System Deployment In order to detect whether a B-scan image I contains traces of objects, we first split I into N_I overlapping patches P_n, $n \in [1, N_I]$ of size $B \times B$. Each patch P_n is fed to the CNN, which associate a vote w_n to each one of them. The idea is that patches extracted from portions of I depicting only background are associated to low w_n values. Conversely, patches containing portions of hyperbola are associated to high w_n values. An example is provided in Fig. 1, which shows that patches located over the hyperbola location are associated by the CNN to a high vote (i.e., yellow in the figure).

After all patches of I are evaluated, we detect the presence of objects by thresholding w_n values. Formally, we associate a label $\hat{l}$ indicating *background* ($\hat{l} = 0$) or *object* ($\hat{l} = 1$) with the following rule:

$$\hat{l} = \begin{cases} 0, & \text{if } \max_n(w_n) < \Gamma \\ 1, & \text{otherwise} \end{cases} \quad , \tag{1}$$

where $\max_n(w_n)$ extracts the maximum value among all w_n, $n \in [1, N_I]$, and Γ is a threshold that can be decided upon a small training set of images.

IV. EXPERIMENTAL SETUP

In this section, we provide details about the considered network architectures, used datasets and experimental methodology.

Network Architectures In order to verify the possibility of using CNNs for buried landmine detection, we tested the proposed pipeline using 3 different network architectures. Architecture $\mathcal{N}_1$ is inspired by the well known LeNet [16], which is composed by 2 convolutional layers with 20 kernels of size 5×5, ReLU activation and 2×2 max-pooling, followed by two fully connected layers of 500 and 2 neurons, respectively. Architecture $\mathcal{N}_2$ is a smaller version of $\mathcal{N}_1$, in which convolutional kernels have been reduced to 3×3 and the number of neurons of the first fully connected layer has been decreased to 250. Finally, $\mathcal{N}_3$ is a version of $\mathcal{N}_1$ with a single convolutional layer, rather than two.

Theoretically, all tested networks accept input patches of small size $B \times B$. In practice, we tested patches for $B \in \{32, 64, 128\}$, corresponding to approximately 8, 16, and 32 cm, respectively. Network training has been performed using stochastic gradient descent with learning rate 0.01 on batches of 64 patches exploiting log-loss activation function for classification. Trained network models have been selected as those minimizing loss on a small validation in the first 10 epochs (i.e., complete passes through all the training images). Training was performed on GeForce GTX 980 GPU, requiring a less than one minute per epoch.

Test Dataset In order to fully validate the proposed system, we strongly believe that real-data from GPR acquisitions must be considered. For this reason, the proposed pipeline has been tested only on real-data. Specifically, real data used in this work were collected using a GPR equipment consisting in an IDS Aladdin (IDS Georadar srl) radar, a shielded ground coupled dipole antenna (spaced 9 cm), with a central frequency and a bandwidth of 2 GHz. A soft pad, the PSG [18], was placed between the radar equipment and the soil to ensure accurate measurements and fixed antenna orientation from trace to trace.

In our setup, we used 9 different targets representing inert landmine models and battlefield debris buried in a sand pit characterized by a very low clay content and a gritty texture, at a depth of approximately 10 cm. We then scanned the area so that each A-scan corresponds to a time window of 20 ns and contains 384 time samples. We obtained 114 B-scans of 180 cm, considering inline sampling of 0.4 cm and crossline sampling of 0.8 cm. By knowing the position of each target, we manually labeled B-scans containing or not object traces. The only processing operations applied to B-scans were automatic resize to match the pixel/cm ratio used by the CNN, and removal of the first few image rows containing the direct path from transmitter to receiver.

Training Dataset An important aspect of the proposed system is the adopted training strategy. As a matter of fact, to effectively train a CNN, datasets of thousands or even millions

(a) Training $\mathcal{N}_1$ on different datasets $\mathcal{D}_R$.   (b) Training $\mathcal{N}_1$ with different patch size B.   (c) Training different CNN architectures.
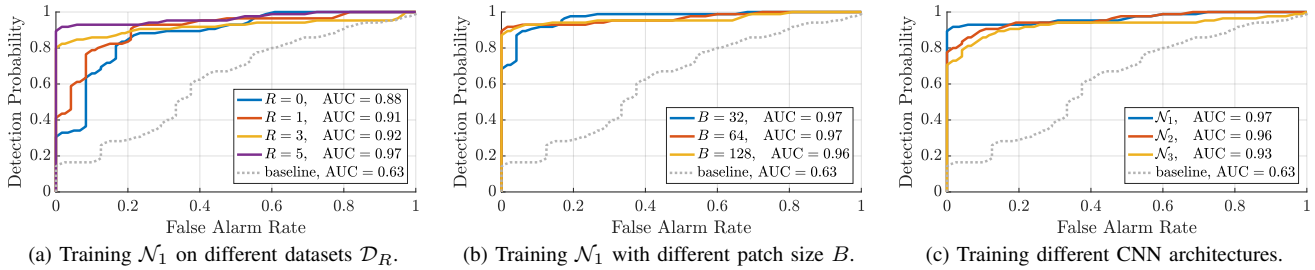
Fig. 2: ROC curves obtained with the proposed solutions: (a) network $\mathcal{N}_1$ for different numbers R of training B-scans from real acquisitions; (b) network $\mathcal{N}_1$ for different patch sizes B; (c) different network configurations $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$, fixing $B = 64$ and $R = 5$.

of images are typically used [14]. This might seem a big issue for the proposed pipeline, as such a huge number of labeled GPR B-scans might not be easily available. However, one of the strong aspects of the proposed architecture is the possibility of being trained on synthetically generated images, still being able to work when deployed on real GPR acquisitions.

To verify this characteristic, we generated 4 different training datasets $\mathcal{D}_R$, where $R \in \{0, 1, 3, 5\}$ indicates the number of real-data B-scans used for training. Specifically, $\mathcal{D}_0$ contains only synthetic patches generated using *gprMax* simulation software [19]. We generated 50.000 background patches and 50.000 patches containing hyperbola portions segmenting simulated B-scans of different ground compositions (i.e., different sands, clays, and compositions) containing or not objects of different shapes (i.e., boxes, spheres and cylinders) with diverse dielectric constants.

Starting from $\mathcal{D}_0$, we then built $\mathcal{D}_1$, $\mathcal{D}_3$ and $\mathcal{D}_5$ by adding to $\mathcal{D}_0$ only background patches from one, three or five B-scans from the real-acquisitions. It is important to notice that the network never uses patches containing real-data hyperbola during training. Therefore, we assume that no information about the target to be detected is available, apart from synthetically generated images.

V. Experimental Results

In this section we present the used evaluation metrics and the achieved results.

Evaluation Metrics As the proposed strategy depends on threshold Γ, we evaluated our method by means of receiver operating characteristic (ROC) curves. These curves represent detection probability and false alarm rate for different values of Γ. Detection probability represents the percentage of B-scans containing an objects correctly detected as such. False alarm rate represents the probability of detecting an object into a B-scan that does not contain it. Good detectors are characterized by ROCs whose area under the curve (AUC) tends to 1. Random guess is characterized by AUC equal to 0.5. As additional metrics we also provide detector accuracy for the best selected Γ.

In order to measure the difficulty of the considered task, we also implemented a simple baseline solution inspired by the pre-screening method used in [12]. Specifically, we computed the average A-scan for each B-scan, and took its maximum value as indicator of high energy returned to the GPR. By thresholding this value we detect the presence of an object. All results are compared to this baseline.

Numerical Results First, we validated the possibility of our system to be trained on synthetic data only, also showing that it is possible to further increase system performance by using some background real-data in the training set. To this purpose, Fig. 2a shows ROC curves obtained using $\mathcal{N}_1$ trained on patches with $B = 64$ from $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_3$ and $\mathcal{D}_5$, respectively. From this figure and numerical results reported in Table I, it is possible to notice that, when only synthetic data are used for training (i.e., $\mathcal{D}_0$), the pipeline still detects buried objects with 83% of accuracy. Moreover, if we add 1 to 5 background B-scans from real-data to the training set, detection accuracy increases up to 95%. Notice that, no matter what dataset is used, the CNN has been never trained using real-data depicting hyperbolic traces. The simple screening baseline only detects objects with 62% of accuracy.

Fig. 3 provides some better insights on the role of using different datasets on landmine detection. Specifically, given a reference B-scan with three targets (Fig. 3(a)), using $\mathcal{D}_0$ for training, the system only detects one target. By increasing the number of real background patches seen during training (Fig. 3(b-e)), the system learns to detect all targets.

After assessing the validity of the training strategy, we tested the effect of using input patches of different size B. Fig. 2b shows results obtained with $\mathcal{N}_1$ architecture trained on $\mathcal{D}_5$ for $B \in \{32, 64, 128\}$. It is possible to notice that results for $B = 32$ and $B = 64$ are barely different. On the other hand, by increasing patch size to $B = 128$, the system experiences a small performance drop. Indeed, big patches capture big portions of hyperbola, thus the CNN does not generalize well enough to hyperbola of slightly different shapes.

Finally, we evaluated the impact of using different network architectures. Fig. 2c show ROC curves obtained using $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ fixing $B = 64$ and using dataset $\mathcal{D}_5$ for training. As a matter of fact, by decreasing CNN size (i.e., $\mathcal{N}_2$ and $\mathcal{N}_3$) a small accuracy decrement is observed.

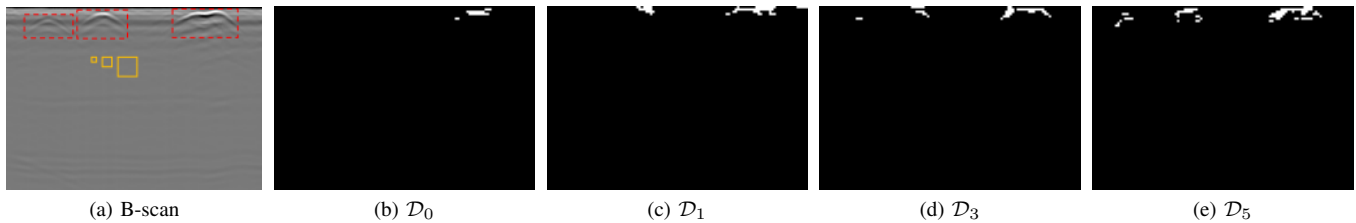Table I reports numeric results in terms of accuracy and AUC for all the presented experiments.

Fig. 3: Examples of detection masks obtained thresholding weights w_n according to Γ. White areas indicate patches detected as containing hyperbolic traces. (a) reference B-scan with three targets (dashed rectangles) and example of patches for $B \in \{32, 64, 128\}$; (b) mask obtained for $R = 0$; (c) mask obtained for $R = 1$; (d) mask obtained for $R = 3$; (e) mask obtained for $R = 5$.

TABLE I: Numerical results for different proposed strategies. Best results in bold, worst in italics.

Algorithm	Patch Size (B)	Dataset (R)	Accuracy	AUC
baseline	-	-	*0.62*	*0.63*
$\mathcal{N}_1$	64	0	0.83	0.88
	64	1	0.85	0.90
	64	3	0.90	0.92
	64	5	**0.95**	**0.97**
	32	5	0.92	**0.97**
	128	5	0.94	0.96
$\mathcal{N}_2$	64	5	0.90	0.96
$\mathcal{N}_3$	64	5	0.88	0.93

VI. CONCLUSIONS

In this paper we presented a pipeline for landmine detection based on the analysis of GPR B-scan images. The proposed approach is based on the use of convolutional neural networks and is fully automated. Validation on real GPR acquisitions show that the system provides up to 95% of accuracy and necessitates of minimal image pre-processing.

Experimental results validated the idea that the CNN can in principle be trained starting from purely synthetic data. However, by adding some background GPR acquisitions to the pool of training images, it is possible to strongly increase detection accuracy. Nonetheless, the system does not need to be trained on images depicting the specific objects of interests from real acquisitions. This characteristic proves paramount for landmine detection scenario. With our approach it is possible to acquire some B-scans of controlled mine-free fields, and deploy the system to detect objects never seen before.

Despite the reported promising performance, the proposed pipeline does not exploit the full capability of the considered GPR system. To further increase accuracy, future work will be devoted to study the effect of using different antenna polarizations. Moreover, we will investigate the possibility of working directly in a 3-dimensional domain, rather than just using B-scans. Finally, we will perform more thorough GPR data acquisition campaigns to study system generalization capability to different kinds of targets.

REFERENCES

[1] International Campaign to Ban Landmines, "Landmine monitor 2015," *Human Rights Watch*, 2015.

[2] T. R. Witten, "Present state of the art in ground-penetrating radars for mine detection," in *SPIE Detection and Remediation Technologies for Mines and Minelike Targets*, 1998.

[3] L. P. Peters, J. J. Daniels, and J. D. Young, "Ground penetrating radar as a subsurface environmental sensing tool," *Proceedings of the IEEE*, vol. 82, no. 12, pp. 1802–1822, 1994.

[4] Y. Liao, L. W. Nolte, and L. M. Collins, "Decision fusion of ground-penetrating radar and metal detector algorithms mdash;a robust approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 2, pp. 398–409, 2007.

[5] D. J. Daniels, "Ground penetrating radar for buried landmine and ied detection," in *Unexploded Ordnance Detection and Mitigation*. Springer, 2009.

[6] U. S. Khan and W. Al-Nuaimy, "Background removal from gpr data using eigenvalues," in *Internarional Conference on Ground Penetrating Radar*, 2010.

[7] D. Carevic, "Clutter reduction and detection of minelike objects in ground penetrating radar data using wavelets," *Subsurface Sensing Technologies and Applications*, vol. 1, no. 1, pp. 101–118, 2000.

[8] A. H. Gunatilaka and B. A. Baertlein, "Subspace decomposition technique to improve gpr imaging of antipersonnel mines," in *SPIE Detection and Remediation Technologies for Mines and Minelike Targets*, 2000.

[9] G. Borgioli, L. Capineri, P. Falorni, S. Matucci, and C. G. Windsor, "The detection of buried pipes from time-of-flight radar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2254–2266, 2008.

[10] H. Chen and A. G. Cohn, "Probabilistic robust hyperbola mixture model for interpreting ground penetrating radar data," in *International Joint Conference on Neural Networks*, 2010.

[11] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k -nearest neighbor classifier," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 185–199, 2009.

[12] P. A. Torrione, K. D. Morton, R. Sakaguchi, and L. M. Collins, "Histograms of oriented gradients for landmine detection in ground-penetrating radar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1539–1550, 2014.

[13] P. D. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden markov models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1231–1244, 2001.

[14] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, January 2009.

[15] L. E. Besaw and P. J. Stimac, "Deep convolutional neural networks for classifying gpr b-scans," in *SPIE Detection and Sensing of Mines, Explosive Objects, and Obscured Targets*, 2015.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," *International Conference on Machine Learning*, 2010.

[18] M. Lualdi, "True 3d acquisition using gpr over small areas: A cost effective solution," in *Symposium on the Application of Geophysics to Engineering and Environmental Problems*, 2011.

[19] C. Warren, A. Giannopoulos, and I. Giannakis, "gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Computer Physics Communications*, vol. 209, pp. 163 – 170, 2016.