Convolutional Autoencoder for Landmine Detection on GPR Scans

Francesco Picetti[1], Giuseppe Testa[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*—Buried unexploded landmines are a serious threat in many countries all over the World. As many landmines are nowadays mostly plastic made, the use of ground penetrating radar (GPR) systems for their detection is gaining the trend. However, despite several techniques have been proposed, a safe automatic solution is far from being at hand. In this paper, we propose a landmine detection method based on convolutional autoencoder applied to B-scans acquired with a GPR. The proposed system leverages an anomaly detection pipeline: the autoencoder learns a description of B-scans clear of landmines, and detects landmine traces as anomalies. In doing so, the autoencoder never uses data containing landmine traces at training time. This allows to avoid making strong assumptions on the kind of landmines to detect, thus paving the way to detection of novel landmine models.**

***Keywords*—Deep Learning; Landmine Detection; GPR**

I. Introduction

The presence of landmines and explosive remnants of war represents a serious threat for civilians around the World. As a matter of fact, even if it is hard to precisely estimate the number of casualties, more than 25.000 people are killed or mutilated every year due to landmines [1]. For this reason, the development of techniques for landmine detection and minefield clearance is of paramount importance.

To implement a complete landmine detection and localization system, a series of different steps have to be developed [2]: (i) *detection* - to detect whether any kind of target is buried within an area of interest, or the area is clear; (ii) *recognition* - to discriminate whether at least one of the buried objects is a landmine, or all objects are just clutter (e.g., stones, wooden sticks, etc.); (iii) *localization* - to determine the precise location of targets of interest. In this work, we focus on the first step, by proposing an automatic system for object detection.

In the literature, many different landmine detection systems have been proposed. Some of them, exploit electromagnetic induction based sensors tailored to capture metal target traces. However, as landmines are nowadays mostly made of plastic, ground penetrating radar (GPR) is emerging as a more suitable technology [3].

A broad family of GPR-based methods works acquiring and analyzing B-scans of the ground, i.e., 2D images in a space-time domain obtained by emitting and recording a signal with a pair of antennas that are moved on a straight line parallel to the ground. B-scans should be ideally flat in case no dielectric discontinuities are present underground. If an object of limited size characterized by a different dielectric constant with respect to the ground is buried (e.g., a landmine), a prominent hyperbola appears. To detect hyperbolas, thus spotting buried objects, different model-based solutions have been proposed. To name a few, [4] solves a fitting problem, [5] proposes a modified Hough transform, whereas [6] and [7] exploit gradient-based features characterizing B-scan texture. Due to the recent astonishing deep-learning advancements in many fields [8], recent methods also started leveraging convolutional neural networks (CNNs) [9], [10].

In this paper, we propose the first landmine detection method leveraging a convolutional autoencoder (i.e., a specific kind of CNN) to analyze B-scans acquired with a GPR. Specifically, we consider the problem of detecting whether a B-scan contains any trace of buried object or not. To do so, we cast landmine detection into an anomaly detection problem, and solve it through a one-class approach. In a nutshell, an autoencoder learns a characterization of B-scans not containing any trace of landmines or other objects at training time. Upon training completion, the autoencoder can be used to detect whether a new B-scan under analysis contains any anomaly with respect to the training set (i.e., presence of hyperbola, thus objects).

The proposed method is completely data driven, but it has the inherent advantage of not making strong assumptions on landmines characteristics (e.g., shape, size, etc.). As long as buried objects introduce some distortion into a B-scan (i.e., hyperbola) compared to B-scans used for training (i.e., obtained from areas without buried landmines), the system is able to identify them. Preliminary results on real GPR data acquired in two different test sites show promising performance compared to a recently proposed method exploiting CNNs [10].

II. Background on Autoencoders

In this section we quickly introduce to the reader the concept of autoencoder needed to understand the rest of the paper. For a thorough autoencoder review, the reader can refer to [11].

This work has been partially supported by the project PoliMIne (Humanitarian Demining GPR System), funded by Polisocial Award from Politecnico di Milano, Milan, Italy.

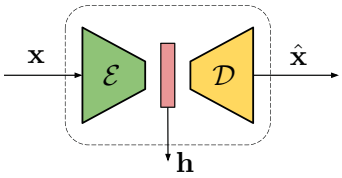


Fig. 1: Scheme of an undercomplete autoencoder. The encoder $\mathcal{E}$ turns the input $\mathbf{x}$ into its hidden representation $\mathbf{h}$. This is turned into $\hat{\mathbf{x}}$ by the decoder $\mathcal{D}$.

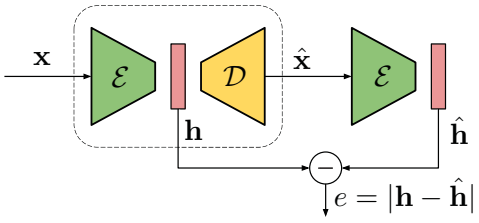


Fig. 2: Diagram of the proposed anomaly detection scheme. A patch under analysis $\mathbf{x}$ is autoencoded to $\hat{\mathbf{x}}$ and encoded again into $\hat{\mathbf{h}}$. Anomaly is detected by thresholding e value.

An *autoencoder* is a specific kind of neural network that takes its name from the ability of being logically split into two separate components: (i) the *encoder*, which is the operator $\mathcal{E}$ mapping the input $\mathbf{x}$ into the so called hidden representation $\mathbf{h} = \mathcal{E}(\mathbf{x})$; (ii) the *decoder*, which is the operator $\mathcal{D}$ that decodes the hidden representation into an estimate of the input $\hat{\mathbf{x}} = \mathcal{D}(\mathbf{h})$. A visual representation of autoencoder is shown in Fig. 1.

In this paper, we refer to a specific family of autoencoders known as *undercomplete convolutional autoencoders*. These are characterized by a hidden representation $\mathbf{h}$ of reduced dimensionality with respect to the input $\mathbf{x}$. Moreover, both encoder and decoder operators are composed by series of linear filtering operations (i.e., convolutions), optionally followed by non linear functions (e.g., sigmoid, hyperbolic tangent, etc.).

By using this kind of autoencoder it is possible to estimate an almost-invertible dimensionality reduction function $\mathcal{E}$ directly from a representative set of training data (i.e., observations of $\mathbf{x}$). A common way of doing this consists in a priori defining a network model (i.e., the series of parametric operations composing $\mathcal{E}$ and $\mathcal{D}$), and estimating the network weights (i.e., the operations' parameters) that minimize some distance metric between the autoencoder input $\mathbf{x}$ and its output $\hat{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$. The used distance metric is typically referred to as loss function, and its minimization is carried out through iterative techniques (e.g., gradient descent methods, etc.). In the light of this, we can interpret the hidden representation $\mathbf{h} = \mathcal{E}(\mathbf{x})$ as a compact feature vector capturing salient information from $\mathbf{x}$.

III. Landmine Detection

In this section we formulate the landmine detection problem faced in this paper, and report all the details about the proposed detection method.

A. Problem

Let us define a B-scan acquired with a GPR system as the 2D image $\mathbf{X}$. If $\mathbf{X}$ has been acquired over a buried target, we associate to it the binary label $l = 1$ indicating the presence of an object underground. If $\mathbf{X}$ has been acquired over a target-free area, we label it with $l = 0$, indicating that no object traces are present. Solving landmine detection problem consists in computing $\hat{l}$ (i.e., an estimate of l) given a B-scan $\mathbf{X}$. Correct detection happens if $\hat{l} = l$. Misclassification happens in case $\hat{l} \neq l$.

B. Proposed Detector

The rationale behind the proposed detector is that autoencoders can be a powerful instrument for anomaly detection [12], [13]. Indeed, an autoencoder tailored to encode and decode a specific kind of data, fails in encoding and decoding correctly other kinds of data. The error introduced in encoded or decoded data can be used as anomaly indicator.

It is therefore possible to train an autoencoder to learn a characteristic hidden representation of B-scans not showing any object traces (i.e., labeled as $l = 0$). After training, this autoencoder will encode and decode B-scans labeled as $l = 0$ with good quality. Conversely, it will encode and decode B-scans labeled as $l = 1$ with poor quality. In the following, we describe each step of the proposed method.

1) System Training: In order to be independent from the B-scan size, we propose to work in a patch-wise fashion. To this purpose, let us consider $\mathbf{x}_i$ as the i-th patch of fixed size extracted from a B-scan $\mathbf{X}$. To train the autoencoder, we define a training set of I patches $\mathbf{x}_i$, $i \in [1, I]$ extracted from B-scans associated to label $l = 0$ (i.e., do not containing any hyperbola due to buried objects). We then estimate the autoencoder weights by minimizing the mean squared error between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ averaged over all patches in the training set.

2) System Deployment: When a B-scan $\mathbf{X}$ is to be analyzed, we split it into a set of I patches $\mathbf{x}_i$, $i \in [1, I]$ covering the whole $\mathbf{X}$. We then follow the block diagram reported in Fig. 2. Each patch $\mathbf{x}_i$ is encoded into $\mathbf{h}_i = \mathcal{E}(\mathbf{x}_i)$. The hidden representation is decoded into $\hat{\mathbf{x}}_i = \mathcal{D}(\mathbf{h}_i)$, which is encoded again into $\hat{\mathbf{h}}_i = \mathcal{E}(\hat{\mathbf{x}}_i)$. We then compare the hidden representation of the original patch (i.e., $\mathbf{h}_i$) with the hidden representation of the autoencoded patch (i.e., $\hat{\mathbf{h}}_i$) by means of Euclidean distance.

The obtained distance $e_i = |\mathbf{h}_i - \hat{\mathbf{h}}_i|$ is an indicator of possible anomalies. Indeed, we expect patches containing hyperbola traces to be incorrectly autoencoded, thus giving rise to $\hat{\mathbf{h}}_i$ strongly different from $\mathbf{h}_i$. Conversely, patches similar to those observed during training should generate $\hat{\mathbf{h}}_i$ very similar to $\mathbf{h}_i$.

To detect landmines, we collect all e_i, $i \in [1, I]$ values belonging to patches coming from the B-scan $\mathbf{X}$ under analysis,

and apply the following criterion

$$\hat{l} = \begin{cases} 1, & \text{if } \max_i(e_i) > \Gamma, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where Γ is a threshold to be selected. In other words, we detect presence of landmines if at least one patch $\mathbf{x}_i$ shows strong evidence of anomaly.

IV. Experimental Setup

In this section we report information about the used network architectures and datasets.

A. Autoencoder Architecture

We tested three different autoencoder architectures to investigate its impact. All architectures are symmetric, as to each convolutional layer used at the encoder, corresponds a deconvolutional layer at the decoder. The input size of each network is equal to its output size. Hidden representations are characterized by a reduced dimensionality with respect to input.

Architecture $\mathcal{N}_1$ is composed by:

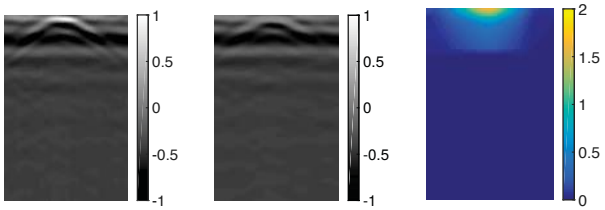
1) A convolutional layer with 16 filters, stride 1x1, size 6x6.
2) Three convolutional layer with 16 filters, stride 2x2, size 5x5, 4x4, 3x3, respectively.
3) A convolutional layer with 8 filters, stride 1x1, size 1x1. Its output is the hidden representation.
4) Four deconvolutional layers with 16 filters, stride 2x2, size 2x2, 3x3, 4x4, 5x5, respectively.
5) A deconvolutional layers with 1 filter, stride 1x1, size 6x6, followed by hyperbolic tangent activation.

This architecture shrinks the input by a factor 32 (e.g., a 32x32 image is turned into a 32 element hidden representation).

Architecture $\mathcal{N}_2$ is the same as $\mathcal{N}_1$, but the convolutional layer returning the hidden representation is substituted by three layers: (i) one convolutional layer with 16 filters, stride 2x2, size 2x2; (ii) one convolutional layer with 16 filters, stride 2x2, size 1x1; (iii) one deconvolutional layer with 16 filters, stride 2x2, size 2x2. This architecture shrinks the input by a factor 64 (e.g., a 32x32 image is turned into a 16 element hidden representation).

Architecture $\mathcal{N}_3$ is the same as $\mathcal{N}_1$, but the convolutional layer returning the hidden representation is substituted by a convolutional layer with 16 filters, stride 2x2, size 2x2. This architecture shrinks the input by a factor 16 (e.g., a 32x32 image is turned into a 64 element hidden representation).

All networks have been trained using Adam optimizer with default parameter until loss function stopped decreasing. Network input was always normalized in range $[-1, 1]$. All tests were run on a workstation equipped with a Titan X GPU reaching convergence in a few minutes.



(a) $\mathbf{X}$ (b) $\hat{\mathbf{X}}$ (c) e_i map

Fig. 3: Example of an original B-scan $\mathbf{X}$ showing an hyperbola (a), its autoencoded version $\hat{\mathbf{X}}$ in which the hyperbola is not perfectly reconstructed (b), and spatial reshape of e_i related to B-scan patches.

B. Dataset

All used data has been acquired using the same system of [10]. Specifically, we used 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 [14], was placed between the radar equipment and the soil to ensure accurate measurements and fixed antenna orientation from trace to trace. We acquired data so that each A-scan corresponds to a time window of 20ns and contains 384 time samples. For B-scans acquisition we considered inline sampling of 0.4cm and crossline sampling of 0.8 cm.

With this system we acquired data from two different test sites. The first setup (i.e., S_1) corresponds to the one presented in [10], consisting of 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. In this setup we acquired 114 B-scans. The second setup (i.e., S_2) consists of 8 targets representing inert landmine models and rocks buried in long jump landing pit sand. In this setup we acquired 64 B-scans. For each setup, we manually labeled each B-scan by knowing where objects were buried.

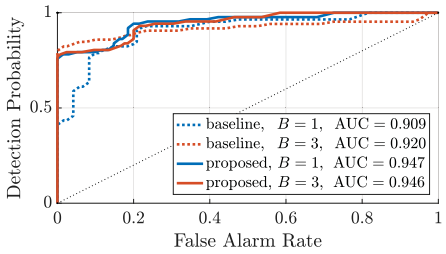
As explained during numerical analysis, we constructed different training datasets by changing the amount of considered training B-scans and setups. For testing, we always considered all B-scans non used for training belonging to setup S_1 only.

V. Experimental Results

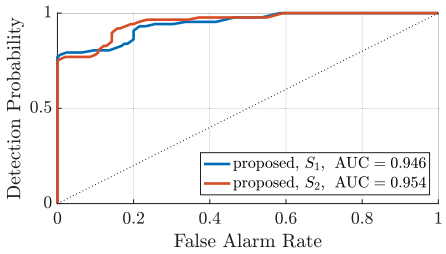
In this section, we explain the used evaluation metrics and collect results from our numerical analysis.

A. Evaluation Metrics

The proposed method is based on a threshold Γ. We therefore evaluated our technique by means of receiver operating characteristic (ROC) curves. A ROC curve represents the probability of correct detection (i.e., correctly finding an object) and probability of false detection (i.e., detecting objects in clear areas) by spanning all possible values of the threshold Γ. This means that each working point of a ROC curve determines a specific Γ value. As compact measure of ROC goodness we selected the area under the curve (AUC). This measure ranges between 0.5 (i.e., random guess) and 1 (i.e., perfect result).



(a) Different training set size B



(b) Different training setup S

Fig. 4: ROC curves under different conditions: (a) proposed $\mathcal{N}_1$ and baseline [10] changing the amount B of used training B-scans; (b) proposed solution trained on setups S_1 and S_2, then tested on S_1.

B. Numerical Analysis

To provide a visual example of the working method, Fig. 3 shows a B-scan region $\mathbf{X}$, its encoded and decoded version $\hat{\mathbf{X}}$, as well as the patch-by-patch error e_i obtained using architecture $\mathcal{N}_1$ with patches of size 32x32. It is possible to notice that the original hyperbola due to a buried object is just only mildly reconstructed in $\hat{\mathbf{X}}$. Conversely, the rest of the B-scan is almost perfectly autoencoded. Computing e_i it is possible to clearly spot an area with high mean square error, corresponding to the detected hyperbola.

To investigate the effect of using more training data, we tested our architecture $\mathcal{N}_1$ using patches of size 32x32 using $B = 1$ or $B = 3$ training B-scans. Fig. 4a shows the ROC curves for our method, and the baseline [10] in the same exact conditions. It is possible to notice that our method improves over [10] when only few data is available for training. Moreover, we apparently do not need to use a high number of training images, as results using $B = 1$ and $B = 3$ are comparable. It is also worth noting that [10] makes some assumptions on the kind of expected hyperbola, as its training contain both patches showing and not hyperbola traces.

Another test we performed consisted in fixing architecture $\mathcal{N}_1$ and changing the image patch size considering 16x16, 32x32 and 64x64. Results remain in line with those presented in Fig. 4a with a maximum AUC deviation of 1%. We therefore stopped our investigation on patch size, considering 32x32 a good choice.

Moreover, we tested the different architectures $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ on setup S_1 using $B = 3$ training images. Also in this case we obtained comparable results, with slight AUC decrease for $\mathcal{N}_2$, which reduces data dimensionality too much. For this reason we decided to only consider $\mathcal{N}_1$ for other tests.

Finally, we performed a cross-dataset test. We trained $\mathcal{N}_1$ on $B = 3$ B-scans split into 32x32 patches using either setup S_1 or S_2. Fig. 4b shows results in terms of testing on setup S_1 only. It is possible to notice that the proposed method is robust against cross-training (i.e., training on S_2 and testing on S_1). This means that the system is not strictly tailored to the only kind of soil used during training.

VI. Conclusions

In this paper we proposed an anomaly detection technique based on convolutional autoencoders for landmine detection in GPR data. The proposed solution is a data-driven approach exploiting a one-class paradigm. Our system uses only data not containing landmine traces at training stage. This makes the system robust to a wide variety of targets, as no strong assumptions are a priori made. Moreover, it is easy to train the system on any specific soil condition. In a practical situation, the system could be trained on a small area that has been previously checked to not contain landmines. Then, it can be used to test neighboring regions. Future work will focus on disambiguation between anomalies due to actual landmines or different buried objects.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] S. Lameri, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro, "Landmine detection from GPR data using convolutional neural networks," in *European Signal Processing Conference (EUSIPCO)*, 2017.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[12] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[13] S. K. Yarlagadda, D. Güera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, "Satellite image forgery detection and localization using GAN and one-class classifier," in *IS&T Electronic Imaging (EI)*, 2018.

[14] 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.