

POLITECNICO DI MILANO
Master in Computer Science Engineering
Dipartimento di Elettronica e Informazione



Landmine Detection on GPR Data Employing Convolutional Autoencoder

Image and Sound Processing Group Lab

Master graduation thesis by:
Giuseppe Testa, matricola 838040

Supervisor:
Prof. Paolo Bestagini
Co-supervisor:
Dott. Francesco Picetti

Academic Year 2017-2018

A mio Padre, che mi ha insegnato a guardare oltre le apparenze.

*A mia Madre, che mi ha trasmesso la forza di affrontare le difficoltà con
il sorriso sulle labbra.*

A Danilo, col cuore.

Abstract

More than ninety nations around the World, which in the past have been a theater of war, have part of their territory characterized by the presence of buried and unexploded landmines. This situation represents severe danger for the civilians living in these areas. Although it is difficult to have an exact count of victims, an estimates sum up to 8000 people either killed or maimed by an explosive remanent of war every year. Humanitarian Demining is regulated by the United Nations (UN) via the International Mine Action Standards (IMAS), that bans the use of military techniques. Military mine clearance is performed with brute force, since they need the infantry to pass through as fast as possible. Sadly, in most of the cases, the only method that meets the constraints imposed by the IMAS consists in manual detection and disarm. This procedure, however, is slow, expensive and dangerous. Among the variously proposed techniques, only a few manage to combine the use of signal processing methods, able to perform accurate detection and classification, with the standard required by the UN. Therefore, advanced methods for on-field use have not been yet implemented. Moreover, the majority of the unexploded ordnances (UXOs) are built almost entirely of plastics and contain a little or no metal. For this reason, in recent years the use of ground-penetrating radar (GPR) have emerged as a suitable sensing modality to acquire images of the subsoil. In the literature, numerous landmine detection systems have been proposed to analyze B-scans (i.e., images acquired by GPR). In particular, the recent Deep Learning advancement in the field of image processing has determinated its use to investigate the mentioned problem.

In this thesis, we propose a methodology for landmine detection on B-scans that makes use of a particular Neural Network, the Autoencoder, that is employed as anomaly detector: the autoencoder learns a description of object-free B-scans, and detects landmines traces as anomalies. It is

worth noticing that the proposed system never uses data containing landmine traces at the training phase. This feature enables us to avoid assumptions about the type of device, therefore allowing to locate any mine independently of its physical and geometrical characteristics. We tested the proposed methodology on real data from on-field acquisitions; the results show an accuracy up to 95% in the recognition of buried objects.

Sommario

Più di novanta paesi del Mondo, teatro di guerra nel passato, si trovano oggi ad avere parte del proprio territorio caratterizzato dalla presenza di mine terrestri, sepolte ed inesplose. Questa situazione rappresenta un pericolo costante per i civili che popolano queste zone. Non si è a conoscenza di un dato certo sulle vittime, ma si stima che circa 8.000 persone ogni anno perdono la vita o vengono mutilate da un ordigno esplosivo. Lo Sminamento Umanitario è regolato dalle Nazioni Unite mediante l'*International Mine Action Standards* (IMAS), il quale proibisce l'utilizzo di tecniche di sminamento militare. Purtroppo nella maggior parte dei casi l'unico metodo che incontra le richieste dell'IMAS consiste nella localizzazione e disarmo manuale. Questo procedimento è però lento, costoso e pericoloso. Le molteplici tecniche proposte non riescono a conciliare l'utilizzo di strumentazioni e/o le tecniche di elaborazione dei dati che permettano una localizzazione e classificazione accurata degli ordigni con le direttive dettate dalle Nazioni Unite. Per questo motivo non esistono ancora metodologie tecnologicamente avanzate che trovino riscontro pratico. Inoltre, molti ordigni sono costruiti quasi esclusivamente in plastica e presentano poco contenuto metallico. Per questa ragione negli ultimi anni l'utilizzo di *ground-penetrating radar* (GPR) come modalità per acquisire immagini del sottosuolo ha visto un notevole incremento del suo utilizzo in questo ambito. In letteratura sono state proposte numerose tecniche per la localizzazione delle mine in *B-scan* (i.e., immagini del sottosuolo acquisite mediante GPR). In particolare, le emergenti tecniche di *Deep Learning* nel campo dell'elaborazione di immagini hanno motivato ad investigare la loro applicazione al problema citato.

In questa tesi, proponiamo un metodo per la localizzazione di ordigni basato su *B-scan* che sfrutta una particolare *Convolutional Neural Network*, l'*Autoencoder*, implementato come rilevatore di anomalie: l'*Autoencoder* apprende una descrizione delle *B-scan* senza ordigni, e rileva un'anomalia

quando processa un'immagine che mostra tracce di mine. È importante sottolineare che il sistema proposto non utilizza dati contenenti tracce di mine nella fase di apprendimento. Questa caratteristica evita ipotesi preliminari sul tipo di ordigno, permettendo di localizzare una mina indipendentemente dalle sue caratteristiche fisiche e geometriche. Attraverso test sul campo abbiamo testato la metodologia proposta ottenendo un'accuratezza del 95%.

Ringraziamenti

Il primo ringraziamento va ai miei genitori, senza di voi non sarei qui. Ma questo lo sapete già. Grazie perchè, anche se spesso non avete compreso o condiviso le mie idee, mi avete sempre supportato ed aiutato. Grazie ai miei cugini, agli zii ed ai nonni tutti.

Un pensiero particolare lo rivolgo al mio paesello, Jelsi, ed a tutti gli amici di una vita perchè ogni volta che torno sembra come se non fossi mai partito.

Se penso la primo giorno al Polimi non posso non pensare alle persone stupende che ho incontrato appena messo piede in terra meneghina, grazie per essermi stati vicini in tutti questi anni.

Vorrei ringraziare un pezzo di Molise con il quale ho condiviso questi ultimi anni *al nord*: i compagni del Calciocavallo FC. Grazie a Giovanni e Francesco per le lunghe chiacchierate e le birrette insieme.

Alla mia seconda famiglia milanese, gli Alcopevoli Inconsalisti, rivolgo un ringraziamento speciale, è difficile trovare persone che ti fanno sentire a proprio agio come lo fate voi con me. In particolare vorrei ringraziare Eddy e Marianna, anche se ci si vede poco ultimamente è difficile dimenticare i viaggi insieme, soprattutto la SEEP Jelsi 2015.

Grazie a Wilmer, per aver reso la mia permanenza a Como meno banale ed a tutte le persone fantastiche che ho incontrato nella SiliComo Valley, soprattutto i coinquilini di Anzani 29.

Di tutte le persone che negli ultimi anni ho perso di vista per i motivi più vari, vorrei ringraziare in particolare Stefano, Fra e Fox. Grazie per avermi dato un'idea di cosa significa non essere figlio unico.

Vorrei ringraziare Paolo per il supporto e l'aiuto ricevuto e Francesco per avermi sopportato negli ultimi mesi.

Infine, l'ultimo pensiero lo dedico alla persona che più di tutte ha condiviso le mie ansie durante questo periodo. Grazie Giulia.

Contents

Abstract	5
Sommario	9
Ringraziamenti	13
1 Introduction	1
1.1 Thesis outline	5
2 Background	7
2.1 Ground Penetrating Radar	7
2.1.1 Basic principles	7
2.1.2 Electromagnetic principles of GPR	9
2.1.3 Physical properties of soil	11
2.2 Artificial Neural Networks	12
2.2.1 Introduction	12
2.2.2 Feedforward Neural Networks	14
2.2.3 Convolutional Neural Networks	19
2.2.4 Autoencoders	22
3 Problem formulation	25
3.1 The landmine detection dilemma	25
3.2 State of the art	27
3.2.1 Model-based approaches	28
3.2.2 Feature-based approaches	29
3.2.3 Deep Learning approaches	29
4 Landmine detection systems	31
4.1 Motivations	31

4.2	Autoencoder overview	32
4.3	The pipeline	33
4.3.1	Dataset and pre-processing	33
4.3.2	System Training	35
4.3.3	System Deployment	35
4.4	Architectures	36
5	Experimental results	41
5.1	Dataset	41
5.1.1	Cranfield	44
5.1.2	Giuriati	45
5.1.3	Synthetic data	46
5.2	Experimental setups	47
5.2.1	Evaluation methodology	48
5.2.2	Baseline	48
5.3	Numerical Results	50
5.3.1	Single dataset approach	50
5.3.2	Cross-dataset approach	54
5.3.3	Results comments	55
6	Conclusions	57
6.1	Future developments	58
	Bibliography	59

List of Figures

2.1	Diagram of GPR system.	8
2.2	A GPR profile obtained over three objects buried in sandy soil. The signal amplitude is plotted as a function of time (or depth) and displacement.	9
2.3	An example of a Feedforward fully connected neural network with two hidden layers	15
2.4	An example of layer connection in a CNN	20
2.5	An example input volume in red, and an example volume of neurons in the first Convolutional layer. Each neuron in the convolutional layer is connected only to a local region in the input.	21
2.6	General structure of an autoencoder.	22
3.1	GPR examples	26
4.1	Detection system pipeline. Training process on top, system deployment on bottom.	33
4.2	Scheme of a complete autoencoder.	34
4.3	Diagram of the proposed anomaly detection scheme for a single patch.	36
4.4	Diagram of the architecture \mathcal{A}_1	37
5.1	Cranfield test site.	42
5.2	Clutter and simulated landmine used in S_2 setup.	43
5.3	Target buried in the Cranfield test site.	44
5.4	Acquisition setup G_2 .	45
5.5	Synthetic patches exemple	46
5.6	Validation Loss vs Epochs.	47

5.7	ROC obtained for different training set size B in comparison	
	to the baseline	48
5.8	GPR examples	50
5.9	MSE vs ground truth	51
5.10	Proposed solution trained on setups S_1 and S_2 , then tested	
	on S_1 .	54

List of Tables

2.1	Typical range of dielectric characteristics of various materials measured at 100 MHz [9]	10
4.1	Architectures \mathcal{A}_2 . The top shows \mathcal{E}_2 (i.e., the encoder), the bottom refers to \mathcal{D}_2 (i.e., the decoder).	38
4.2	Architectures \mathcal{A}_3	39
5.1	Impact of different patch parameters on architecture \mathcal{A}_1 , trained with $B = 5$ B-scans of \mathcal{S}_1 dataset.	53
5.2	Impact of the different numbers of B-scans for training on dataset of \mathcal{S}_1 , patch size 64×64 , and stride 4×4	53
5.3	Best results of cross-tests between \mathcal{S}_1 and \mathcal{S}_2	55

Chapter 1

Introduction

In 2016, an average of 23 people around the world lost their life or limb to unexploded ordnance (UXO) or another explosive remanent of war (ERW), every day [18]. That means over 8 thousand people were hurt or killed only in one year. Landmines contaminate about 90 countries and areas around the world and thousands of people continue living with this daily threat of losing their lives. This reality explains why the development of techniques for landmine detection is of paramount importance.

The challenge of minefield clearance has been tackled over the year in a lot of different ways. Nevertheless, we can outline some tasks that are common to the various techniques. To implement a complete landmines detection and localization system, a series of different steps have to be developed [33]:

- detection - to detect whether any target is buried within an area of interest, or the area is clear;
- recognition - to discriminate whether at least one of the detected object is a landmine, or all the objects are just clutter (e.g., stones, wooden sticks, etc.);
- localization - to determine the precise location of targets of interest.

Among the technique developed in the literature, some of them exploit electromagnetic induction (EMI) based sensors tailored to capture metal target

traces. Unfortunately, nowadays the majority of UXO are made of plastics and contain little or no metal, making the use of EMI inappropriate for this purpose. For this reason, ground-penetrating radars (GPR) started to be incorporated in most of the new minefield clearance systems thanks to its sensitivity to plastic materials [22] [21].

Ground-penetrating radar is a geophysical method that uses radar pulses to image the subsurface. A transmitter emits electromagnetic energy into the ground. When the wave encounters a buried object or a boundary between materials having different permittivities, it may be reflected or refracted or scattered back to the surface. A receiving antenna can then record the variations in the return signal. Moving the two antennas on a straight line parallel to the ground while recording the signal, the system provides a 2D image in a space-time domain named B-scans. The image should be ideally flat in case of homogeneous soil with no dielectric discontinuities (e.g., the case of purely sandy soil). 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. The detection challenge consists in deciding whether a B-scan contain a hyperbola (i.e., if a buried object is present in the analyzed image).

Since the first applications of GPR, the scientific community studied its application in different fields. In the task of buried objects detection, we can find in the literature many GPR signal and image processing techniques. These methods have a common general schematics: 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 or background subtraction [16] [8]. Then, the processed data is examined to detect the presence of buried targets.

To detect hyperbolas, thus spotting buried objects, both model-based detection methods and features-based solutions have been proposed. To name a few, [10] solves a fitting problem, [6] proposes a modified Hough transform, whereas [15] and [31] exploit gradient-based features characterizing B-scan texture. Due to the recent astonishing Deep Learning advancements in many fields [3], recent methods also started leveraging Convolutional Neural Networks (CNNs) [5], [17].

In the last years, a new field of Machine Learning called Deep Learning had been proved to be an excellent tool with respect to more classic ML technique. The breakthrough of Deep Learning had a substantial impact on Machine Learning and in particular on Computer Vision, drastically advancing the state of the art of Image Understanding algorithms. In particular, the task of Image Recognition (i.e., the ability of software to identify objects, places, people, writing and actions in images) achieved, with the advent of Deep Learning era, performances that exceed those obtained with more classical approaches. Different from a classic ML paradigm, the DL techniques are focused on providing a way to learn a feature representation automatically. The idea behind Deep Learning is indeed to learn a general feature representation that can be exploited for different tasks instead of building hand-engineered features that rely on fixed heuristics. This can be done, for example, by stacking several layers of Neural Networks, each layer learns a more complex representation of the data. For instance, Convolutional Neural Networks trained on images learn similar levels of representations as of the human brain where the first layer learns simple edge filters, the second layer captures primitive shapes and higher levels combine these to form objects. The fantastic successes obtained with CNN in the field of Image Recognition and the lack of applications in the field of minefield clearance led us to investigate if good results can be achieved.

The goal of this thesis is to develop a method to solve the task of buried object detection in GPR B-scans (i.e., 2D images of vertical underground slices) based on the use of Convolutional Neural Networks (CNNs).

The first step was to collect GPR acquisition to perform our investigations. The first dataset used was the same as [17] and it was composed of images acquired in a sand pit characterized by a very low clay content and a gritty texture during a study conducted in the UK. A second dataset was built with GPR data captured on the long-jump pit of the university campus: the sand is a homogenous soil (i.e., noiseless), it is relatively easy to bury and dig up objects, and reflects a typical situation of the contaminated ground over the world. Moreover, even if seems similar to the other dataset, some difference due to the different conditions of the sites results in very different images. Having two different set of data allowed us to investigate

the performances of our method with a cross-dataset approach.

Once in posses of sufficient data, we develop a learning-based algorithm. We reverse the typical paradigm using a data-driven methodology that learns features that characterizes buried targets directly from GPR images, rather than imposing any model or hand-crafted feature recipe, The algorithm was trained over a particular kind of CNN architectures: the Autoencoder. This technique has been proved to be a powerful instrument for anomaly detections. We can, therefore, imagine the hyperbolas present within B-scan as the anomaly we want to detect and train the autoencoder to learns a characterization of B-scans not containing any trace of landmines or other objects. The system, trained in this way, can be used to predict whether a new B-scan contain or not the anomaly (i.e., an object). The main advantage of this paradigm is the lack of necessity of GPR data containing threats. This characteristic brings two significant benefits: using only non-contaminated underground acquisitions gives to the system a relatively easy way of taking into account the properties of different soils; on the other hand, we do not need to make strong assumptions on landmines characteristics (e.g., shape, size, etc.). Because 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.). We also understood the importance, for future use in real applications, of making the pipeline as independent as possible from any manipulation of the input. Therefore we tried to strip down the image pre-processing phase (i.e., only track synchronization, removal of the direct antennas path and normalization). Moreover, the autoencoder overcomes one of the primary constraints in DL: the needs of a significant amount of data to train a Neural Network. This can be a significant problem since the procedure for GPR data acquisition is very time expensive, making infeasible the creation of a database with enough data.

For a robust system, we need to consider the possibility of having acquisitions coming from different areas (i.e., with different dielectric properties). To do so, we used a cross-dataset approach: the GPR acquisitions from a single test site where used for training a CNN while the data from the other test site where used to test the performance of the architectures. With this method as long as the 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 can identify them.

We tested the proposed methodology through field tests on real data. The results shows that using only a few B-scans we can reach up an accuracy of up to 95% in the recognition of buried objects.

The results obtained in this thesis were collected in the article [25], published in 41st IEEE Conference on Telecommunications and Signal Processing (TSP), held in Athens, Greece, 4/6 July 2018.

1.1 Thesis outline

The thesis is structured as follows.

In Chapter 2 we provide a theoretical background on GPR and Neural Networks, with a focus on a specific application of CNN called Autoencoder.

In Chapter 3 we formalize the problem and discuss the issues and the requirements that are involved. Furthermore, we report the state-of-the-art for the landmines detection problem.

In Chapter 4 we detail the implementation of the system built to solve the landmines dilemma. We report the pipeline used as well as the detail about the architectural choices.

In Chapter 5 we describe the performed experiments, giving an interpretation of the obtained results. Furthermore, we illustrate the construction of the datasets.

In Chapter 6 we illustrate our conclusions on the presented work, and we discuss some possible future developments and improvements.

Chapter 2

Background

This Chapter reviews the theoretical background that is behind our investigation. Section [2.1](#) describes the instrument used to acquire the images analyzed in this study: the ground penetrating radar (GPR). For a complete review of GPR and its application refer to [\[13\]](#). Section [2.2](#) gives a theoretical background of the machine learning techniques employed to build the system.

2.1 Ground Penetrating Radar

The ground-penetrating radar (GPR), is a non-destructive geophysical technique that uses radar pulses to image the subsurface [\[22\]](#). It employs electromagnetic radiation in the microwave band of the radio spectrum and detects the reflected signals from subsurface structures. Since its first applications in the 1660s, there have been rapid developments in hardware, measurement and analysis techniques, and the method has been extensively used in many applications, such as archaeology, forensics, geology, and buried explosive hazards (BEHs) detection [\[14\]](#).

2.1.1 Basic principles

The GPR directs a pulse of electromagnetic radiation into the ground and measures the return signal's amplitude as a function of time (or depth),

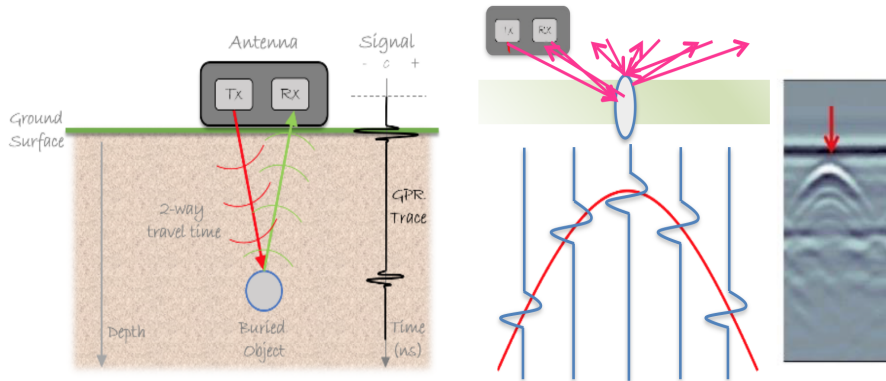


Figure 2.1: Diagram of GPR system.

called an A-scan. Individual A-scans characterize a narrow slice of the subsurface. The structure of an A-scan is strongly affected by the medium through which the radiation propagates. If the medium contains regions with vastly different dielectric constants, it may be reflected or refracted or scattered back to the surface. A receiving antenna can then record the variations in the return signal. Consequently, the A-scan will exhibit complex reflections at the interfaces where the dielectric constant changes. Groups of successive A-scans are referred to as B-scans and provide a more effective means for characterizing and visualizing subsurface phenomena. This procedure is illustrated in Figure 2.1. To obtain a B-scan, the system scans the ground to collect the data at various locations. Then a GPR profile can be constructed by plotting the amplitude of the received signals as a function of time and position, representing a vertical slice of the subsurface, as shown in Figure 2.2. The time axis can be converted to depth by assuming a velocity for the electromagnetic wave in the subsurface soil.

The electrical conductivity of the ground, the transmitted center frequency, and the radiated power may limit the effective depth range of GPR investigation. Increases in electrical conductivity attenuate the introduced electromagnetic wave, thus the penetration depth decreases. Because of frequency-dependent attenuation mechanisms, higher frequencies do not penetrate as far as lower frequencies. However, higher rates may provide improved resolution. Therefore, the choice of the operating frequency is

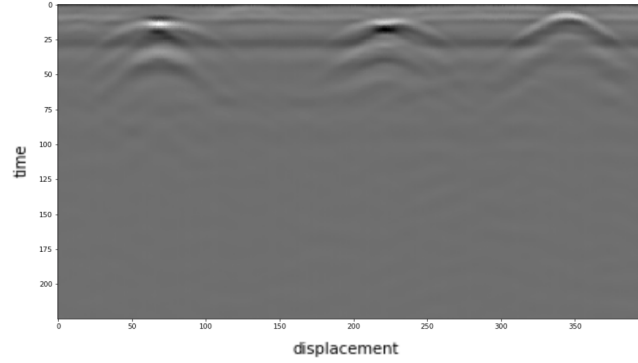


Figure 2.2: A GPR profile obtained over three objects buried in sandy soil. The signal amplitude is plotted as a function of time (or depth) and displacement.

always driven by a trade-off between resolution and penetration.

2.1.2 Electromagnetic principles of GPR

This Section illustrates the electromagnetic principle underlying the functioning of the GPR.

Electromagnetic wave propagation in soil

The propagation velocity v of the electromagnetic wave in soil is characterized by the dielectric permittivity ε and magnetic permeability μ of the medium:

$$v = \frac{1}{\sqrt{\mu\varepsilon}} = \frac{1}{\sqrt{\mu_0\mu_r\varepsilon_0\varepsilon_r}} \quad (2.1)$$

where $\varepsilon_0 = 8.854 \times 10^{-12}$ F/m is the permittivity of free space, $\varepsilon_r = \varepsilon/\varepsilon_0$ is the relative permittivity (dielectric constant) of the medium, $\mu_0 = 4\pi \times 10^{-7}$ H/m is the free-space magnetic permeability, and $\mu_r = \mu/\mu_0$ is the relative magnetic permeability. In most soils, magnetic properties are negligible, yielding $\mu = \mu_0$, and Equation (2.1) becomes

$$v = \frac{c}{\sqrt{\varepsilon_r}} \quad (2.2)$$

Table 2.1: Typical range of dielectric characteristics of various materials measured at 100 MHz [9].

Material	Relative permittivity [S/m]	Conductivity
Air	1	0
Freshwater	81	$10^{-6} - 10^{-21}$
Clay, dry	2-6	$10^{-3} - 10^{-1}$
Clay, wet	5-40	$10^{-1} - 10^{-0}$
Sand, dry	2-6	$10^{-7} - 10^{-3}$
Sand, wet	10-30	$10^{-3} - 10^{-2}$

where $c = 3 \times 10^8$ m/s is the speed of light.

In general, dielectric permittivity ε and electric conductivity σ are complex and can be expressed as

$$\varepsilon = \varepsilon' - j\varepsilon'' \quad (2.3)$$

$$\sigma = \sigma' - j\sigma'' \quad (2.4)$$

where ε' is the dielectric polarisation term, ε'' represents the energy loss due to the polarisation lag, σ' refers to ohmic conduction, and σ'' is related to faradaic diffusion. Table 2.1 provides the typical range of permittivity, conductivity and attenuation of various materials.

Reflection and transmission of waves

GPR methods usually measure reflected or scattered electromagnetic signals from changes in the electric properties of materials. When electromagnetic waves impinge upon a planar dielectric boundary, some energy is reflected at the boundary and the remainder is transmitted into the second medium. The relationships of the incident reflected, and transmitted electric field strengths are given by

$$E^i = E^r - E^t \quad (2.5)$$

$$E^r = R \cdot E^i \quad (2.6)$$

$$E^t = T \cdot E^i \quad (2.7)$$

respectively, where R is the reflection coefficient, and T is the transmission coefficient. In the case of normal incidence the reflection and transmission coefficients are given as

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (2.8)$$

$$T = 1 - R = \frac{2Z_2}{Z_2 + Z_1} \quad (2.9)$$

where Z_1 and Z_2 are the intrinsic impedances of the first and second media, respectively, and $Z = \sqrt{\mu/\varepsilon}$. In a low-loss non-conducting medium, the reflection coefficient may be simplified as [13]

$$R = \frac{\sqrt{\varepsilon_{r1}} - \sqrt{\varepsilon_{r2}}}{\sqrt{\varepsilon_{r1}} + \sqrt{\varepsilon_{r2}}} \quad (2.10)$$

2.1.3 Physical properties of soil

As seen in the previous section, the electric and magnetic properties of a medium influence the propagation and reflection of electromagnetic waves. These properties are dielectric permittivity, electric conductivity, and magnetic permeability.

Dielectric permittivity

Permittivity describes the ability of a material to store and release electromagnetic energy in the form of electric charge and is classically related to the storage ability of capacitors. Permittivity greatly influences the electromagnetic wave propagation in terms of velocity, intrinsic impedance, and reflectivity. The relative permittivity (dielectric constant) of air is 1, is between 2.7 and 10 for common minerals in soils and rocks, while water has a relative permittivity of 81, depending on the temperature and frequency. Thus, the permittivity of water-bearing soil is strongly influenced by its water content.

Electric conductivity

Electric conductivity describes the ability of a material to pass free electric charges under the influence of an applied field. The primary effect of conductivity on electromagnetic waves is an energy loss, which is expressed as

the real part of the conductivity. The imaginary part contributes to energy storage and the effect is usually much less than that of energy loss. In highly conductive materials, the electromagnetic energy is lost as heat and thus the electromagnetic waves cannot propagate as deeply. Therefore, GPR is ineffective in materials such as those under saline conditions or with high clay contents. At GPR frequencies, the conductivity is often approximated as real-valued static or DC conductivity.

Magnetic permeability

The magnetic property of soils is caused by the presence of ferrimagnetic minerals, mainly magnetite, titanomagnetite, and maghemite. These minerals either stem from the parent rocks or can be formed during soil genesis. As discussed in previous sections, the magnetic properties theoretically influence the propagation of electromagnetic waves. However, in natural soils, the influence of the magnetic properties of the soil is fairly low in most cases. The magnetic permeability must be extremely high to influence the GPR signal. Therefore, the magnetic permeability of most soils is usually assumed to be the same as that of free space, i.e., $\mu = \mu_0$ and $\mu_0 = 1$.

2.2 Artificial Neural Networks

This Section provides the theoretical background of Deep Learning useful to understand the rest of the thesis. After a brief introduction, we begin by describing the model that is at the foundation of the Neural Network design: the Feedforward Neural Network. Later, we discuss more complicated models called Convolutional Neural Networks that are widely used in Computer Vision tasks. Finally, we detail a particular application of CNNs that is the core of our system: the Autoencoder.

2.2.1 Introduction

Machine Learning (ML) lies within the field of Artificial Intelligence (i.e., the study of machines able to mimics functions that require human intelligence, such as visual perception, speech recognition, decision-making) and it refers

specifically to *the capacity of a computer to learn from experience, i.e. to modify its processing on the basis of newly acquired information.* [26]

To better understand this definition we make use of a fine description of the concept of learning given by Mitchell [24]: *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .* In this definition, an example refers to a collection of features that have been quantitatively measured from some event that we want the machine learning system to process (e.g., the features of an image are usually the values of the pixels in the image). The experience is (within) a collection of examples, called dataset. Starting from the experience, we define the tasks in terms of how the machine learning system should process an example. Machine learning has been proved during the years to be an excellent medium for solving many kinds of tasks, such as classification, anomaly detection, regression, etc. The abilities of a machine learning algorithm are evaluated by its performance with some quantitative measure P , specific to the task T being carried out by the system. For example, the task of classification is measured by the accuracy of the model (i.e., the proportion of examples for which the model produces the correct output). Normally, we are interested in how well the machine learning algorithm performs on data that it has not seen before since this determines how well it will work when deployed in the real world. We, therefore, evaluate these performance measures using a test set of data that is separate from the data used for training the machine learning system.

Machine Learning algorithms can be categorized into two broad categories depending on the on the knowledge available during the learning process:

- Supervised learning: the data are associated with a label, provided by an instructor or teacher who shows the machine learning system what to do.
- Unsupervised learning: no labels are associated with the input features.

In this thesis, we will focus on the supervised learning paradigm, particularly

in a family of Machine Learning models called Artificial Neural Networks (ANN). ANN, also called Neural Networks (NN), are methods that allow a machine to automatically discover the representations of a given row data input to perform tasks such as detection or classification. These methods compose simple non-linear modules that, starting from the raw data, each transforms the representation at one level into a representation at a higher, more abstract level. In this way can learn very complex functions [19]. Formalizing the Mitchell definition, the object of a neural network algorithm is to estimate the best parameters which approximate a given function minimizing the error between the examples and the approximation given by the model (i.e., minimizing a predefined performance measure).

Before discussing more complicated and specialized models such as Convolutional Networks and Autoencoders, the next chapter presents a brief discussion of the basics of the ANNs.

2.2.2 Feedforward Neural Networks

The modern Artificial Neural Networks sow the seeds in the model of the artificial neuron, proposed by Rosenblatt with the name of Perceptron [28]. The Perceptron was the peak of the numerous researches conducted during the 50s with the aim of modeling biological neural systems. As defined by Rosenblatt, the Perceptron simply consists in a supervised binary classifier. The investigation of this method terminated rapidly after it was proved that could not be trained to recognize many classes of patterns. Years later, it was recognized that more layers of Perceptron can be stacked together and trained with the backpropagation algorithm [30], from these ideas born the Feedforward Neural Network (also called a Multilayer Perceptron). Despite the biological interpretation, we can think the feedforward network as a method to approximate some function f^* . The term feedforward refers to the topology of the approximation function: information flows from the input x , through intermediate computations, used to define f , to the output y . The name networks come from the fact that their representation is typically given by the composition of many different functions and the composition of this functions can be described with a graph.

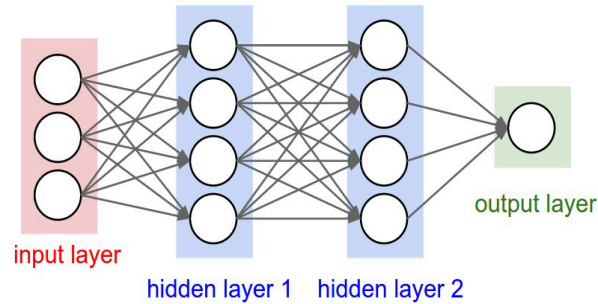


Figure 2.3: An example of a Feedforward fully connected neural network with two hidden layers

Neural Network models are organized into distinct layers of many units that act in parallel representing a vector to a scalar function. Each unit resembles a neuron in the sense that it receives input from many other units and computes its own activation value. Let us take for example three functions $f^{(1)}$, $f^{(2)}$, $f^{(3)}$ connected in a chain form $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$. This structure resembles a typical neural network, where each function represents a layer of the architecture. The term deep came from the multiple functions that form the chain. The final layer of the network is called the output layer. The neural network transforms the input through a series of hidden layers into a new representation that is more suitable for the task to be performed by the system. During the training, we drive $f(\mathbf{x})$ to match $f^*(\mathbf{x})$. Each noisy example provided by the training is associated with a label. The training examples specify the desired output that each input point must produce. However, the behavior of the other layers is not directly specified by the training data. The learning algorithm must decide how to use the intermediate layers to produce the desired output but the training data does not say anything on what each layer should do. Since the training data does not show the desired output for each of the intermediate layers, these are called hidden layers.

Training a Neural Network

The art of designing a Neural Networks requires careful hyperparameters decision. In general, we have to choose:

- The optimizer
- The cost function
- The hidden units
- The output units
- The activation functions
- The topology of the network

The actual training of the model follows the design phase. Specifically, once we have picked all the hyperparameters, we perform a gradient-based optimization that iteratively adjusts the model parameters to drive the cost function (i.e., the relationship between the output and the model's parametrization) to the global minimum.

Unfortunately, non-linearities and high dimensionality cause the cost function to be highly non-convex and make the optimization very hard in practice. For this reasons, many successful investigations have proposed various alternatives of the classic gradient descent algorithm, that consists in refinements of the well known stochastic gradient descent (SGD) algorithm. The SGD algorithm is an approximation of the classic gradient descent algorithm that uses an estimate of the gradient of the loss function based only on a single example of the training set.

In order to minimize the cost, we need to compute the gradient of the cost function with respect to the models' parameters. To compute the gradient of the cost function, thus performing the descent towards its minimum, the best current method is given by the backpropagation algorithm. Once the information flows forward from the input layer passing through the hidden units and producing the output prediction \hat{y} , the training algorithm iteratively updates the parameters of the network based on the cost function $J(\theta)$ evaluation, where θ indicates all network's parameters to be learned

(i.e., biases b and weights w). The backpropagation algorithm allows the information given by the cost to flow backward through the network in order to compute the gradients.

The learning of the weights is obtained by propagating the prediction error back through all the layers of the network it is possible. At each step, the algorithm updates each weight of a quantity proportional to the gradient of the error function with respect to the weight. A general updating rule for the weights of the network can be written as follows:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial J(\theta)}{\partial \mathbf{w}} \quad (2.11)$$

where w is the vector of weights and α is the learning rate. Since the output is given by a composition of non-linear functions it is sufficient to apply several times the classic chain rule to compute the gradients. To give a brief description of the chain rule, let x be a real number, and consider the composite function $z = f(g(x)) = f(y)$, with $y = g(x)$, of two functions $g, f : \mathbb{R} \rightarrow \mathbb{R}$. Then, the derivative of f with respect to x can be computed by applying the chain rule as follows:

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \quad (2.12)$$

Once the gradient of the cost function is computed, the gradients are used to perform the parameters update.

A peculiarity of the FNN, with respect to the classic ML algorithms, is the presence of the hidden units. The first proposals have been the logistic sigmoid activation function

$$g(x) = \sigma(x) \quad (2.13)$$

and the hyperbolic tangent activation function

$$g(x) = \tanh(x). \quad (2.14)$$

However, sigmoidal units saturate to a high value when x is very positive and saturate to a low value when x is very negative. This makes the gradient-

based learning very hard. For this reason, they have been substituted by other kinds of hidden units in feedforward networks. Nowadays they are mostly used as output units or in other settings like autoencoders.

For FNN, the most popular activation function is the Rectified Linear Unit (ReLU). The ReLU activation function is linear with slope 1 for positive arguments and zeroes otherwise, i.e., uses the following activation function

$$g(x) = \max\{0, x\}. \quad (2.15)$$

Typically a rectified hidden unit is used on top of an affine transformation:

$$\mathbf{h} = g(\mathbf{W}^T \mathbf{x} + \mathbf{b}). \quad (2.16)$$

We have seen that a feedforward neural network provides several hidden layers defined by $\mathbf{h} = f(\mathbf{x}; \theta)$. The output layer provides a last additional transformation to complete the task the network must perform. Generally, any kind of hidden units can be used also as output. One of the simplest kind of output is the linear unit. Given features \mathbf{h} , a layer of linear output units produces a vector $\hat{\mathbf{y}} = \mathbf{W}^T \mathbf{h} + \mathbf{b}$. Linear output layers are often used to produce the mean of a conditional Gaussian distribution. In a maximum likelihood framework maximizing the log-likelihood is equivalent to minimizing the mean squared error.

A different approach, used for example in the binary classification tasks, is the sigmoid output units combined with maximum likelihood.

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{h} + b) \quad (2.17)$$

We can think of the sigmoid output unit as having two components. First, it uses a linear layer to compute $z = \mathbf{w}^T \mathbf{h} + b$. Next, it uses the sigmoid activation function to convert z into a probability.

If we want to represent a discrete probability distribution over k possible outcome, a possible solution uses a generalization of the sigmoid function: the Softmax function. Softmax functions are used as the output of a classifier to normalize in the probability range. In this case, we need to produce a vector $\hat{\mathbf{y}}$, with $\hat{y}_i = P(y = i|\mathbf{x})$, where each element \hat{y}_i need to be between 0

and 1 and all sum to 1 in order to represent a valid probability distribution. First, a linear layer is used to predict unnormalized log probabilities

$$z = \mathbf{W}_T \mathbf{h} + \mathbf{b}, \quad (2.18)$$

where $z_i = \log \tilde{P}(y = 1 | \mathbf{x})$. The softmax function can then exponentiate and normalize \mathbf{z} to obtain the desired output

2.2.3 Convolutional Neural Networks

Feedforward Neural Nets don't scale well when working with images as inputs. For example, an RGB image of size 32×32 after a single fully-connected neuron in regular Neural Network would have at least $32 \times 32 \times 3 = 3072$ weights (i.e., without considering the bias and the number of neurons within the layer). This amount still seems manageable but is obvious that this fully-connected structure does not scale to larger images. Clearly, this full connectivity is wasteful and the huge number of parameters would quickly lead to overfitting. A way to overcome this problem is given by Convolutional Neural Networks (CNNs or ConvNets) [20]. CNNs are a specialized type of neural network for processing data that has a known grid-like topology. ConvNet architectures make use of linear filters (i.e., convolutions) instead of just sums and products between matrices. This property becomes really valuable when the inputs are images (i.e., a 2D grid of pixels).

Taking advantage of the fact that the input consists of images the CNNs constrain the architecture in a more sensible way. In particular, as shown in Figure 2.4 the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth. The neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner.

The CNN layer's parameters consist of a set of learnable filters, spatially small (along width and height), but extends through the full depth of the input volume.

In the case of 2D inputs, we convolve each filter across the width and height of the input volume during the forward pass, and compute dot products between the entries of the filter and the input at any position. As we

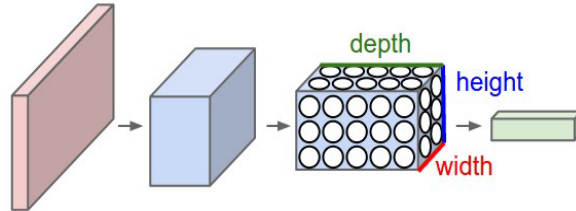


Figure 2.4: An example of layer connection in a CNN

slide the filter over the width and height of the input volume we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual features such as an edge of some orientation or a blotch of some color on the first layer. Each filter produces a separate 2-dimensional activation map, that are stacked along the depth dimension and produce the output volume. The output is controlled by three hyperparameters: the depth, stride, and zero-padding. The depth of the output corresponds to the number of filters we use, each learning to look for something different in the input. We also must specify the stride with which we slide the filter, i.e. the intent of which we move the filters along the image. A stride higher than one will produce spatially smaller output volumes. Moreover, sometimes it will be convenient to pad the input volume with zeros around the border. Zero padding will allow controlling the spatial size of the output volumes. Finally, the spatial size of the output volume can be computed as a function of the input volume size, the receptive field size of the Conv Layer neurons, the stride with which they are applied, and the amount of zero padding used on the border.

By way of conclusion, it worth to review three important ideas that make convolutions improving the neural network system:

Local connectivity is the property of CNN of connecting each neuron to only a local region of the input volume. The spatial extent of this

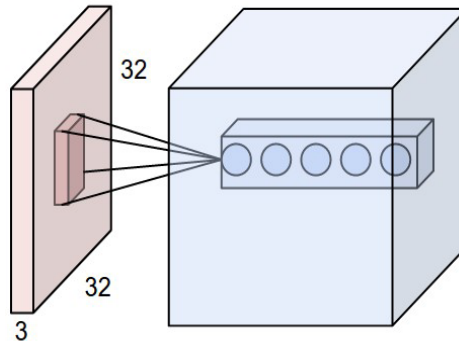


Figure 2.5: An example input volume in red, and an example volume of neurons in the first Convolutional layer. Each neuron in the convolutional layer is connected only to a local region in the input.

connectivity is a hyperparameter called the receptive field of the neuron (equivalently this is the filter size). The space between each receptive field is called stride. Figure 2.5 provides a visual representation of this characteristic.

Parameter sharing refers to the use of the same parameter for more than one function in a model. It turns out that we can dramatically reduce the number of parameters by making this assumption. In practice during backpropagation, every neuron in the volume will compute the gradient for its weights, but these gradients will be added up across each depth slice and only update a single set of weights per slice. Notice that if all neurons in a single depth slice are using the same weight vector, then the forward pass of the convolutional layer can in each depth slice be computed as a convolution of the neuron's weights with the input volume. This is why it is common to refer to the sets of weights as a filter (or a kernel), that is convolved with the input.

In the case of convolution, parameter sharing causes the layer to have an important property for images that is called **equivariance** to translation. The convolution creates a map of where certain features appear in the input. Moving the object in the input, its representation will move the same amount in the output. For example, it is useful that the network learns a robust

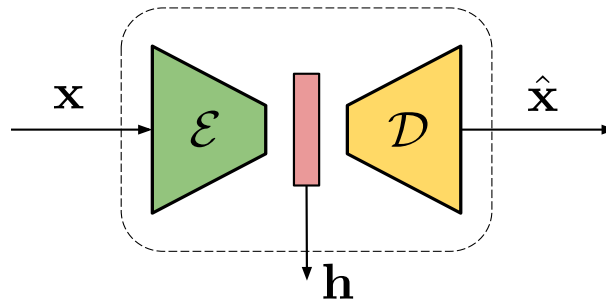


Figure 2.6: General structure of an autoencoder.

edge detector because the same edges appear more or less everywhere in the image. Notice that, convolution is not naturally equivariant to some other transformations, such as changes in the scale or rotation of an image. Other mechanisms are necessary for handling these kinds of transformations.

2.2.4 Autoencoders

An autoencoder is a specific kind of neural network that is trained to obtain an output very close to its input. In doing so, it creates a hidden layer \mathbf{h} that describes a code used to represent the input. It takes its name from the ability to be 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 Figure 2.6.

Undercomplete Autoencoders

The interesting ability of the autoencoder does not lie in simply copy the input to the output. Instead, we training the autoencoder to perform the input copying task provide us with useful properties of the input.

A way to achieve this result is to constrain \mathbf{h} to have a smaller dimension than \mathbf{x} . This forces the autoencoder to capture the most salient features of the training data. An autoencoder whose code dimension is less than the input dimension is known as undercomplete autoencoder. Moreover, both encoder and decoder operators are composed by a series of linear filtering

operations, optionally followed by non-linear functions (e.g., sigmoid, hyperbolic tangent, etc.). When the filtering operations include one or more convolutional layers the autoencoder is said convolutional undercomplete autoencoder.

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 $\mathbf{x} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$. The learning process is described simply as minimizing a loss function

$$L(\mathbf{x}, \mathcal{D}(\mathcal{E}(\mathbf{x}))) \tag{2.19}$$

where L is a loss function penalizing $\mathcal{D}(\mathcal{E}(\mathbf{x}))$ for being dissimilar from \mathbf{x} , such as the mean squared error. Typically, the minimization of L is carried out through iterative techniques (e.g., gradient descent methods, etc.). In the light of this, we can interpret the hidden representation $\mathbf{h} = \mathcal{D}(\mathbf{h})$ as a compact feature vector capturing salient information.

Chapter 3

Problem formulation

In this Chapter, we formulate the landmine detection problem focusing on its issues and constraints. Moreover, in Section [3.2](#), we cover the state-of-the-art of landmines detection.

3.1 The landmine detection dilemma

An adequate solution to the problem posed by landmines implies that the percentage of detected mines in the area under analysis should approach 100%. Furthermore, the detection task needs to be performed at the fastest rate possible and with the lowest false alarms possible (i.e., mistaking a clutter object for a mine). To set a reference, the United Nations has set the detection goal at 99.6%, and the U.S. Army's allowable false-alarm rate is one false alarm in every 1.25 square meters. No existing landmine detection system meets these criteria and the reasons for this failure can be conferred to the mines themselves and the variety of environments in which they are buried as to the limits or flaws in the current technology [\[2\]](#).

The very first issue of any landmines detection system is to obtain some knowledge about the underground. In our work, we choose GPR acquisitions (Chapter [2.1](#)) to gather information about the subsurface. GPR is a non-destructive geophysical technique used for a variety of shallow subsurface imaging applications that provides an effective means for characterizing and visualizing subsurface phenomena. Besides, GPR has proved to be an im-

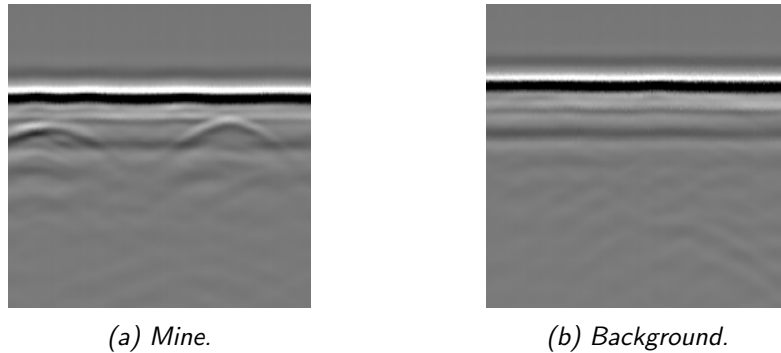


Figure 3.1: Examples of two GPR acquisitions.

portant sensing modality for detecting most types of BEHs [22]. As shown in Figure 3.1, B-scans present characteristic hyperbolic traces when GPR analyze profiles over buried objects (a). Conversely, if the ground is relatively object-free, B-scans do not show prominent hyperbolas (b).

Landmines are found in a variety of soils and terrain: rocky or sandy soil, open fields, forested areas, steep terrain, jungle. A suitable detection system must be able to locate all types of mines individually in a variety of environments. A valid system should account for the differences due to the different characteristics of the underground.

Figure 3.1 shows a high amplitude white band occurring roughly at depth bin 50. This band corresponds to the point at which the radiated electromagnetic waves cross the air-ground interface. Over very uneven or rough terrain, this ground bounce can complicate the interpretation of GPR data, leading to missed targets and false alarms. Good practice suggests the removal of the white band in the pre-processing phase [4].

Generally, landmines are buried as close to the surface as possible, and the detonation is typically caused by pressure. Thus, a landmine detector must do its job without having direct contact with mine or consider to locate a barrier between the surface and the radar.

We can now reformulate the problem in the following way. Given a B-scan (i.e., a map of the underground), it is possible to determine if contains, with some probability, a trace of a buried object? Can our system accurately discriminate from an image containing a mine from a landmine-free acquisition?

Formally, let us define a B-scan acquired with a GPR system as the 2D image \mathbf{X} . The detection system receives \mathbf{X} as input and, if \mathbf{X} corresponds to an acquisition took over a buried target, it associates the binary label $l = 1$ to \mathbf{X} , indicating the presence of an object underground. If \mathbf{X} has been acquired over a target-free area, it labels \mathbf{X} with $l = 0$, indicating that no object traces are present. Solving the landmine detection problem consists in computing \hat{l} (i.e., an estimate of l) given a B-scan \mathbf{X} . It worth to notice that correct detection happens if $\hat{l} = l$, and misclassification happens in case $\hat{l} \neq l$.

It is worth to notice that working on B-scans sequentially (i.e., processing the images in order as we are physically moving along the contaminated area) provides a lower bound for the algorithm performances: to reach 100% detection is enough to identify an object in at least one of the B-scans in which appears. Giving the relevance of this application, since the life of the deminer is at stake, having such bound it is of a paramount importance to avoid False Positive detection (i.e., do not detect a landmine when it's should).

3.2 State of the art

The aforementioned problem has been faced over the last year trying to take advantage of the rapidly increasing technological innovation. In the literature, many GPR signal and image processing techniques have been proposed to automatic detect patterns associated with buried objects (i.e., hyperbolas), thus spotting landmines.

Generally, these methods first implement a data pre-processing step that performs tasks as data normalization, correction for variations in depth and speed, removal of stationary effects due to the system response, background subtraction [7] [16]. Then, processed data is analyzed to extract some features representative of hyperbolic traces or fit a geometric model. As the last step, a classification algorithm is applied to detect the presence of buried targets. To this purpose, both model-based detection methods and features-based techniques have been proposed. Moreover, due to the recent astonishing Deep Learning advancements in many fields [3], recent methods

also started leveraging Artificial Neural Networks (ANNs), particularly on a specific class of ANN called Convolutional Neural Network (CNN), most commonly applied to analyzing visual imagery [17] [5].

3.2.1 Model-based approaches

Model-based approaches for Object Recognition try to represent (approximate) the object as a collection of three dimensional, geometrical primitives (boxes, spheres, cones, cylinders, generalized cylinders, surface of revolution) or by its shape/contour [29]. Specifically, for the task of landmine detection, typical model-based approaches aim to individuate hyperbolic shapes in GPR images.

In [10], Chen proposes a probabilistic robust hyperbola mixture model based on classification Expectation Maximization (EM) algorithm. This method incorporates a hyperbola fitting into the probabilistic method, specifically, a geometric model, in order to account for the hyperbolic shape caused by a landmine into a B-scan. On this model, it applies the EM, in which, at each step estimate the hyperbola parameters based on orthogonal distance, and maximize the likelihood based on the founded estimate. The classification is made by choosing the point that maximizes the posterior probability.

A different approach to spot the geometric shapes provoked by the presence of an object into a radar acquisition is the Hough transform. The Hough transform is a feature extraction technique used in image analysis to find imperfect instances of objects within a certain class of shapes by a voting procedure in the parameter space. These methods aim to identify the four parameters related to the hyperbola, facilitating the subsequent estimation of the buried assets. In [6], Borgioli presents a method relying on the Ridge Detection method. He proposes an improvement of the classic method by introducing a weighting factor that enables optimally placed sets of data pairs to be given greater weight than ill-conditioned sets (e.g., when all data pairs lie near one end of the arc). The major drawback of this approach is that requires to specify a suitable threshold for the number of votes to determine the number of hyperbolae in the image.

A major problem to this approaches is the sensitivity of GPR systems to changes in local environmental conditions results in highly variable responses from buried objects that hinder correct hyperbola detection. To overcome this problem, a series of feature-based techniques were developed.

3.2.2 Feature-based approaches

Detection algorithms based on statistical feature extraction looks for feasible matches between object features and image features, extracting features from the objects to be recognized and the images to be searched (e.g., surface patches, corners, linear edges, etc.). The derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

In [15], Frigui defines an algorithm for landmine detection system that uses edge histogram descriptors for feature extraction and a possibilistic K-nearest neighbors (K-NNs) rule for confidence assignment. In His work, first uses a prescreening algorithm for anomaly detection to identify the candidate mines. The identified regions are processed by a feature extraction algorithm to capture their salient features. Next, the training signatures are clustered to identify prototypes. The main idea is to identify a few prototypes that can capture the variations of the signatures within each class. These variations could be due to different mine types, different soil conditions, different weather conditions, etc. Fuzzy memberships are assigned to these representatives to capture their degree of sharing among the mines and false alarm classes. Finally, a possibilistic K-NN-based rule is used to assign a confidence value to distinguish true detections from false alarms.

3.2.3 Deep Learning approaches

During the last years, the golden age of Deep Learning inspired numerous reasearch groups to quest for new applications of this tecniques in the most different areas. The landmine detection problem was also taken into consideration to investigate if the results obtained with DL, and in particular with Convolutional Neural Networks, were able to exceed the ones obtained

with more classical ML and Signal Processing approaches.

In [5], Besaw presents a method that leverage on a CNN algorithm applied on B-scans. The processing pipeline starts with a pre-processing, such as spatial resampling, ground-bounce tracking and alignment, and A-scan phase alignment. After the first phase, a Deep Belief Network is used as a prescreening (anomaly detection) followed by a series of post-processing steps to identify anomalous signatures. GPR B-scans are extracted from each of these anomalies and used to train and evaluate the BEH discrimination algorithms. After extracting the B-scans, Stimac applies a 2D median filter over them as well as a zeros scores component analysis (ZCA) technique to smooth and whiten the GPR B-scans. Finally, the obtained B-scans is fed to the CNN discriminating algorithm that performs classification based on a threshold.

Another successful result in this way was carried out by Lameri et al. in [17]. The classification pipeline designed by Lameri includes a classic CNN model and proved that a two-class approach gives surprising results in the task of landmine detection. The system was tailored to work on a dataset composed of synthetic data together with real acquisitions, where each image was tagged with a label stating whether or not it contains mine traces. The CNN was fed with a portion of the images (i.e., the training dataset) and their label and outputs the learned model. Once the CNN performed its training, it can be used to classify the images of the test dataset (i.e., images never used in the training step). Specifically, it is possible to feed the CNN with an unlabeled B-scan and obtain a vote proportional to the likelihood of containing a portion of a hyperbola. The higher the vote, the more likely the analyzed image has been captured over a landmine. The module to perform the actual classification of a new image is build on top of the CNN. Specifically, in order to detect whether a B-scan image contains traces of objects, the system first splits the B-scan into a number of overlapping patches, then, each patch is fed to the trained model, which associates a vote to each one of them. After all the patches are evaluated, the presence of an object is detected by thresholding the votes: if a patch within the image obtains a vote bigger than the threshold, then a landmine is detected, otherwise it is classified as mine-free.

Chapter 4

Landmine detection systems

In this Chapter, we present a comprehensive description of our method for landmines detection. After illustrating the motivations behind our choices, Section [4.2](#) recalls the salient characteristics of the autoencoder, that is the heart of our system. Section [4.3](#) provides a detailed explanation of our pipeline. Finally, Section [4.4](#) shows in detail the neural network architectures used in the system.

4.1 Motivations

We choose to address the task of identifying a hyperbola within a given acquisition with a peculiar neural network paradigm: the Autoencoder. The rationale behind this choice is that autoencoders can be a powerful instrument for anomaly detection [\[12\]](#). In the task of anomaly detection, a program sifts through a set of events or objects and flags some of them as being atypical. For our purpose, we can consider the simple background image as the data we want to analyze and the hyperbola as the anomaly.

Behind this motivation, some other strong considerations support this approach to tackle the landmine detection problem:

1. to train the system, we do not need to acquire data of real landmines (avoiding the danger of the operation);
2. from a theoretical point of view, since we do not consider any repre-

sentations of the mines within our model, we do not need to make any assumptions about the UXO structure.

4.2 Autoencoder overview

Any classification system that includes a Neural Network needs to be analyzed into two phases: first, the network must be tuned in order to learn a set of parameters useful to solve the problem under analysis. This is done by training the model over a portion of the data in the possession. The second phase is the actual deployment of the detector: the trained system receives a new image as input and predicts the probability of the input belonging to any class..

The classic two-class approach consists in training over a dataset composed of data (i.e., images) that can be divided into two categories based on their peculiarities. To do so, each image within the dataset requires a tag that indicates the class to which it corresponds. We can see our scenario as a two-class supervised problem considering the B-scans coming from different class whether or not they contain a trace of prominent hyperbolas. Notwithstanding our system seems to resemble a two-class method, in practice we do not train our system with images of both classes, instead, we let the system learn only one class.

Specifically, as we have seen in Section [2.2.4](#) the Autoencoder is a type of Artificial Neural Network with a peculiar property. This specific NN tries to generate an output very close to its input, but, in this process, it creates a hidden representation $\hat{\mathbf{h}}$ of the input; if we force $\hat{\mathbf{h}}$ to have a smaller dimension than the input, the Autoencoder will capture, within the hidden representation, the most salient features of the training data. The process of reducing the dimensionality of a digital object is known as encoding. 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. Therefore, it is possible to train an autoencoder to learn a hidden symbolic representation of B-scans not showing any object traces. After training, the Autoencoder will encode and decode B-scans of pure background soil (i.e.,

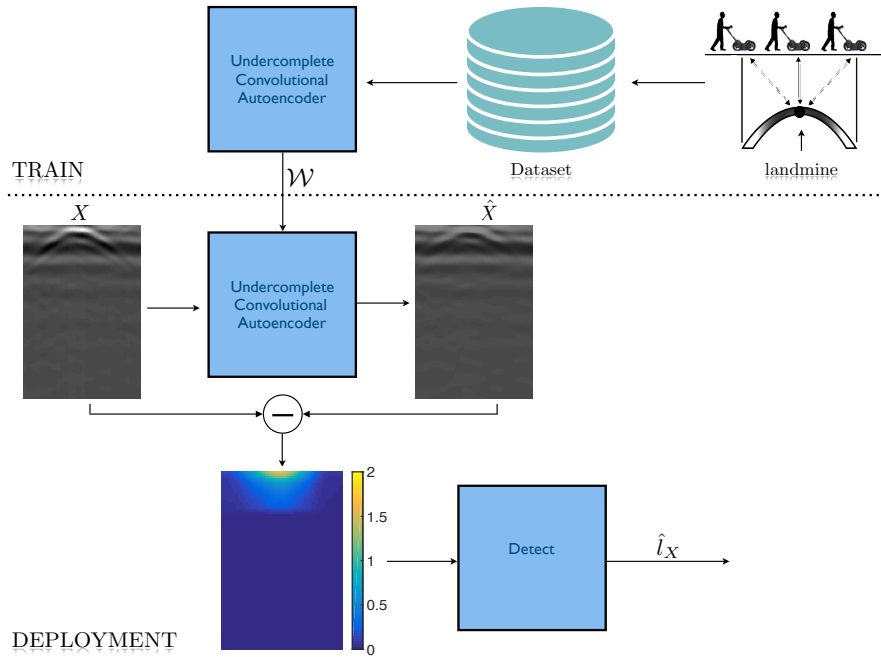


Figure 4.1: Detection system pipeline. Training process on top, system deployment on bottom.

images captured over a landmines-free area) with a small error. Conversely, it will encode and decode B-scans containing prominent hyperbolic traces with poor quality. Measuring the performance, with some error measure, we can detect where the system is failing to indicate that the image presents an anomaly (i.e., an object).

4.3 The pipeline

To perform classification on a new B-scan based on anomaly detection we build a series of modules that compose our landmine detection system. In the following, we outline the proposed pipeline, depicted in Figure 4.1, detailing each module.

4.3.1 Dataset and pre-processing

The preliminary step to our work was to create a dataset of images. For a detailed description of the dataset used in this work and its generation,

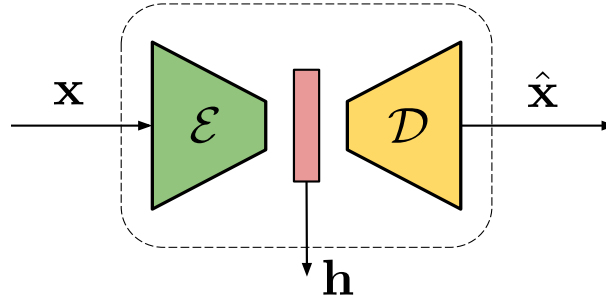


Figure 4.2: Scheme of a complete autoencoder.

refer to the next Section (5.1). For each experiment, we start from an initial dataset composed of M B-scans. Let us define the images belonging to this dataset as \mathbf{X}_j , where $j \in [1, M]$. Each image is associated with a label l_n depending on its category: background images (i.e., containing only background noise) are labeled with $l_n = 0$ while acquisitions containing portions of a hyperbola are labeled with $l_n = 1$. Our system is tailored to work with small patches extracted from the of real data. Therefore, each image \mathbf{X}_j is split into $B \times B$ overlapped patches \mathbf{x}_n . We decided to work in a patch-wise fashion to be independent of the B-scan size. To our purpose, let us consider \mathbf{x}_n^j as the n -th patch extracted from a B-scan \mathbf{X}_j . The dataset was split into training, validation and test dataset, respectively \mathcal{D}_{train} , \mathcal{D}_{val} and \mathcal{D}_{test} . For the training and validation, we make use only of images labeled with $l_n = 0$ (i.e., pure background). While the test was performed on images of both classes.

We would like to emphasize the importance of working with small images (i.e., small patches obtained by cropping the original acquisition). The purposes of this choice are different: first, it makes the system independent of the original acquisition size; besides, growing the size of the training dataset results in better training of the network.

The training stage needs to receive harmonious input data. To respect this constraint, the data were pre-processed bringing the dynamic range of real images within the same interval $[-1, 1]$.

4.3.2 System Training

Given a training dataset \mathcal{D}_{train} and a NN architecture \mathcal{A} , the subsequent step is to learn the set of parameters \mathcal{W} (i.e., filter coefficients, inner product weights, etc.) that characterize the dataset features.

The training module is composed of two steps: the encoder and the decoder. The patches \mathbf{x}_n from our training dataset are fed into the encoder producing their hidden representation \mathbf{h}_n . In the second step, \mathbf{h}_n are given to the decoder producing the reconstructed patch $\hat{\mathbf{x}}_n$. We then estimate the autoencoder weights by minimizing the mean squared error between \mathbf{x}_n and $\hat{\mathbf{x}}_n$ averaged over all patches in the training set. In this way, we impose the system to learn how to encode patches extracted from the images of our dataset properly. As a consequence, the system will perform poorly in the task of decoding images that differs from the ones that have seen. Therefore, once the autoencoder is trained, it can be used to classify (i.e., estimate l) the patches of the test dataset (i.e., images never used in the training step). Specifically, it is possible to feed the autoencoder with an unlabeled patch \mathbf{x}_n and obtain a vote w_n (i.e., the difference between the patch and its decoded version) proportional to the likelihood of \mathbf{x}_n to contain an anomaly and thus derive the membership class .

4.3.3 System Deployment

When a new B-scan \mathbf{X} (i.e., extracted from the test dataset) is to be analyzed, we split it into a set of N_X patches \mathbf{x}_n , $n \in [1, N_X]$ covering the whole \mathbf{X} . We then follow the block diagram reported in Figure [4.3](#). Each patch \mathbf{x}_n is encoded into $\mathbf{h}_n = \mathcal{E}(\mathbf{x}_n)$. The hidden representation is decoded into $\hat{\mathbf{x}}_n = \mathcal{D}(\mathbf{h}_n)$, which is encoded again into $\hat{\mathbf{h}}_n = \mathcal{E}(\hat{\mathbf{x}}_n)$. We then compare the hidden representation of the original patch (i.e., \mathbf{h}_n) with the hidden representation of the auto-encoded patch (i.e., $\hat{\mathbf{h}}_n$) through Euclidean distance. The obtained distance $e_n = |\mathbf{h}_n - \hat{\mathbf{h}}_n|$ is an indicator of possible anomalies. Indeed, we expect patches containing hyperbola traces to be incorrectly auto-encoded, thus giving rise to $\hat{\mathbf{h}}_n$ strongly different from \mathbf{h}_n . Conversely, patches similar to those observed during training should generate e_n close to zero.

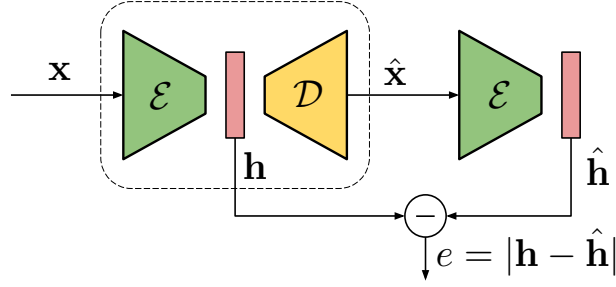


Figure 4.3: Diagram of the proposed anomaly detection scheme for a single patch

To detect landmines, we collect all e_n , $n \in [1, N_X]$, values belonging to patches coming from the B-scan \mathbf{X} under analysis, and apply the following criterion:

$$\hat{l}_X = \begin{cases} 1, & \text{if } \max_n(e_n) < \Gamma \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

where Γ is a threshold to be selected. In other words, we detect the presence of landmines if exist at least one patch \mathbf{x}_n , for some n , that shows substantial evidence of anomaly.

4.4 Architectures

In this work, we proposed different architectures to determine if a variation of the model complexity, regarding the number of layers, produce a significant effect on the performance. 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 a B (i.e., the size of the patch) and it is equal to its output size. Hidden representations are characterized by a reduced dimensionality as to the input. The convolutional and deconvolutional layers are labeled respectively as C_i and D_i , where i is the layer index.

The first architecture, called \mathcal{A}_1 and illustrated in Figure 4.4 is composed of the following ten layers.

- Layer C_1 is a convolutional layer with 16 feature maps. Each unit in each feature map is connected to a 6×6 neighborhood in the input,

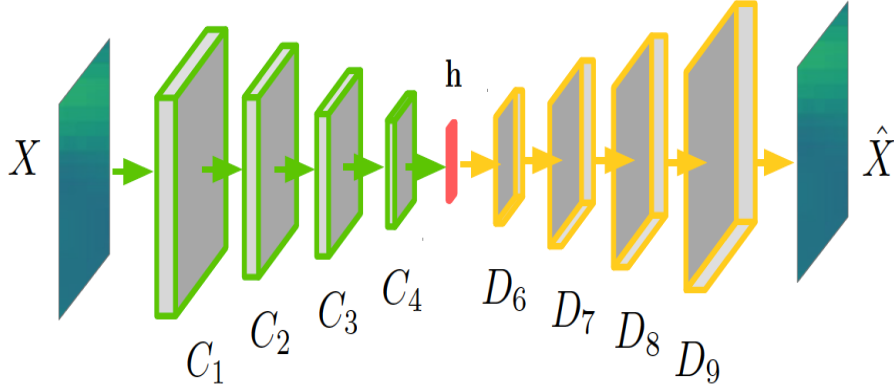


Figure 4.4: Diagram of the architecture \mathcal{A}_1

and it has a 1×1 stride.

- Layer C_2 is a convolutional layer with 16 feature maps, stride 2×2 and size 5×5 .
- Layer C_3 is a convolutional layer with 16 feature maps, stride 2×2 and size 4×4 .
- Layer C_4 is a convolutional layer with 16 feature maps, stride 2×2 and size 3×3 .
- Layer C_5 is a convolutional layer with 6 feature maps, stride 1×1 and size 1×1 . The output of the fifth layer is the hidden representation.
- Layers D_6, D_7, D_8 and D_9 are four deconvolutional layers with 16 feature maps, stride 2×2 and size $2 \times 2, 3 \times 3, 4 \times 4, 5 \times 5$, respectively.
- Layer D_{10} is a deconvolutional layer with one filter, stride 1×1 , size 6×6 , followed by hyperbolic tangent activation.

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

Architecture \mathcal{A}_2 is the same as \mathcal{A}_1 , but the convolutional layer returning the hidden representation is substituted by three layers: (i) one convolutional layer with 16 filters, stride 2×2 , size 2×2 ; (ii) one convolutional layer with 16 filters, stride 2×2 , size 1×1 ; (iii) one deconvolutional layer with 16 filters, stride 2×2 , size 2×2 . This architecture shrinks the input

Table 4.1: Architectures \mathcal{A}_2 . The top shows \mathcal{E}_2 (i.e., the encoder), the bottom refers to \mathcal{D}_2 (i.e., the decoder).

	Layer	Kernel size	Stride	Num. filters	Output size
\mathcal{E}_2	C_1	6×6	1×1	16	$B - 5 \times B - 5 \times 16$
	C_2	5×5	2×2	16	$B - 9 \times B - 9 \times 32$
	C_3	4×4	2×2	16	$B - 12 \times B - 12 \times 48$
	C_4	3×3	2×2	16	$B - 14 \times B - 14 \times 64$
	C_5	2×2	2×2	16	$B - 15 \times B - 15 \times 80$
	C_6	1×1	2×2	16	$B - 15 \times B - 15 \times 96$
\mathcal{D}_2	D_7	2×2	2×2	16	$B - 15 \times B - 15 \times 80$
	D_8	2×2	2×2	16	$B - 14 \times B - 14 \times 64$
	D_9	3×3	2×2	16	$B - 12 \times B - 12 \times 48$
	D_{10}	4×4	2×2	16	$B - 9 \times B - 9 \times 32$
	D_{11}	5×5	2×2	16	$B - 5 \times B - 5 \times 16$
	D_{12}	6×6	2×2	1	$B \times B$

by a factor of 64 (e.g., a 32×32 image is turned into a 16 element hidden representation). Table 4.1 detail the layers of this architecture.

Architecture \mathcal{A}_3 , depicted in Table 4.2, is the same as \mathcal{A}_1 , but the convolutional layer returning the hidden representation is substituted by a convolutional layer with 16 filters, stride 2×2 , size 2×2 . This architecture shrinks the input by a factor of 16 (e.g., a 32×32 image is turned into a 64 element hidden representation).

Table 4.2: Architectures \mathcal{A}_3 .

	Layer	Kernel size	Stride	Num. filters	Output size
\mathcal{E}_3	C_1	6×6	1×1	16	$B - 5 \times B - 5 \times 16$
	C_2	5×5	2×2	16	$B - 9 \times B - 9 \times 32$
	C_3	4×4	2×2	16	$B - 12 \times B - 12 \times 48$
	C_4	3×3	2×2	16	$B - 14 \times B - 14 \times 64$
	C_5	2×2	2×2	16	$B - 15 \times B - 15 \times 80$
\mathcal{D}_3	D_6	2×2	2×2	16	$B - 15 \times B - 15 \times 80$
	D_7	3×3	2×2	16	$B - 12 \times B - 12 \times 48$
	D_8	4×4	2×2	16	$B - 9 \times B - 9 \times 32$
	D_9	5×5	2×2	16	$B - 5 \times B - 5 \times 16$
	D_{10}	6×6	2×2	1	$B \times B$

Chapter 5

Experimental results

In this Chapter, we present the experimental setups and the results achieved in this work. Specifically, Section 5.1 shows a detailed report of the datasets used for the experiments, as well as their construction. In Section 5.2, we illustrate the setup common to all the experiments carried out in this work. In Section 5.3, we give a detailed explanation of our results in comparison to recently developed state-of-the-art solutions.

5.1 Dataset

The dataset employed for our investigation consists of a set of B-scan images (i.e., real data acquisitions made by a Ground Penetrating Radar, described in Section 2.1). Specifically, the data 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 [23], was placed between the radar equipment and the soil to ensure accurate measurements and fixed antenna orientation from trace to trace.

We made use of two different datasets collected at different sites. We made this choice moved by the conviction that a robust system must be as independent as possible from the data used during the training. To prove the strength of our model we use a cross-dataset approach in which we perform the train on a dataset and the test on the other.

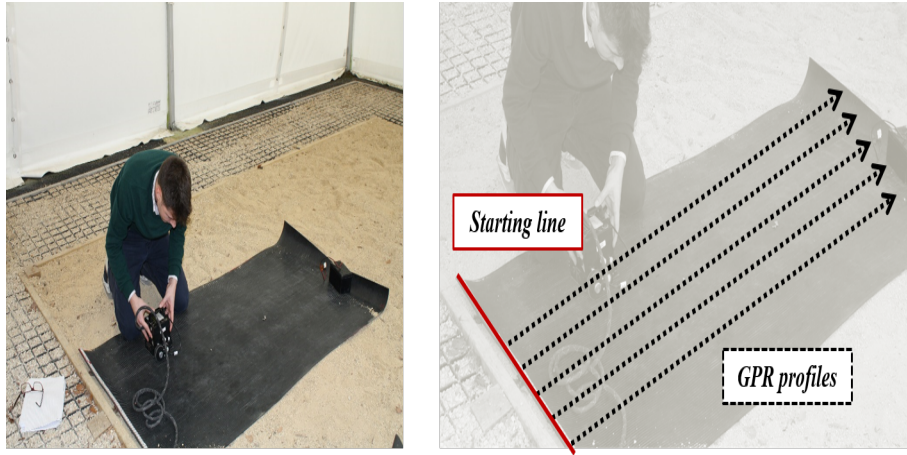


Figure 5.1: Cranfield test site.

The first dataset analyzed was the same of [17] and consist of acquisitions performed in a military camp base in the UK, captured for university investigations purposes. The second was assembled inside the PoliMi campus trying to reproduce a contaminated area with different objects. The test sites were chosen to be representative of different situations, in terms of objects, that could be encountered in a real landmines contaminated area. To this purpose, in both setups were used different targets representing inert landmine models and battlefield debris buried at a depth of approximately 10cm to resemble a realistic situation. We acquired different sets of data on both the sites. After the acquisitions, we manually labeled each B-scan by knowing where the objects were buried for each setup. This procedure gives approximately half of B-scans containing object traces. The following subsections describe in detail the two test sites.

Landmines



Clutter & UXO

□



Figure 5.2: Clutter and simulated landmine used in S_2 setup.

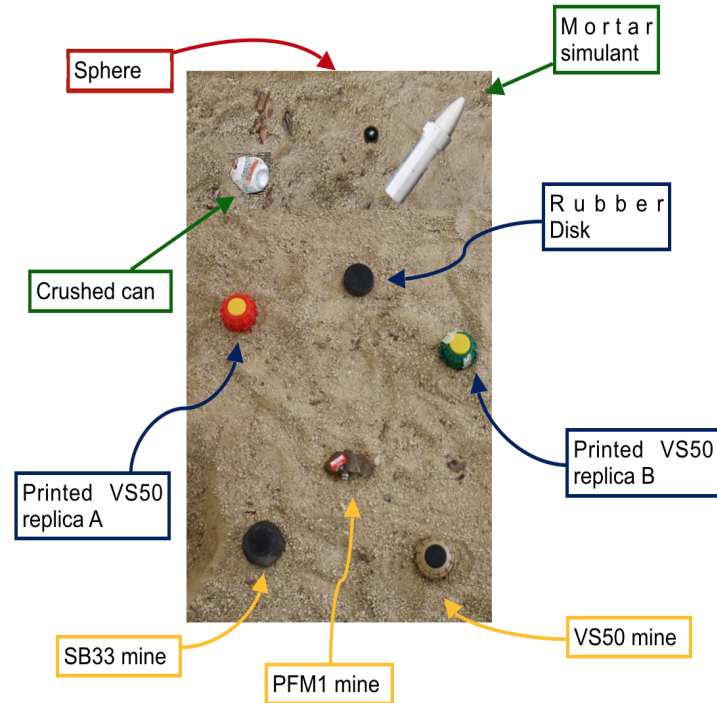


Figure 5.3: Target buried in the Cranfield test site.

5.1.1 Cranfield

The first dataset, called Cranfield and indicated with S_1 , was built in a controlled site at a military base in the UK during February 2017. The soil of the sand pit in which were buried the objects was characterized by a very low clay content and a gritty texture; the humidity due to the rainy weather during the time of acquisitions imply a high content of water within the soil. Figure 5.1 shows the positioning of the PSG over the area under analysis and illustrate the starting line of the acquired GPR profiles.

In this setup, were used 9 different targets (Figure 5.3) representing inert landmine models and battlefield debris, at a depth of approximately 10cm. The area was scanned so that each A-scan corresponds to a time window of 20 ns and contains 384 time samples, obtaining 114 B-scans of 180 cm, considering an 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.

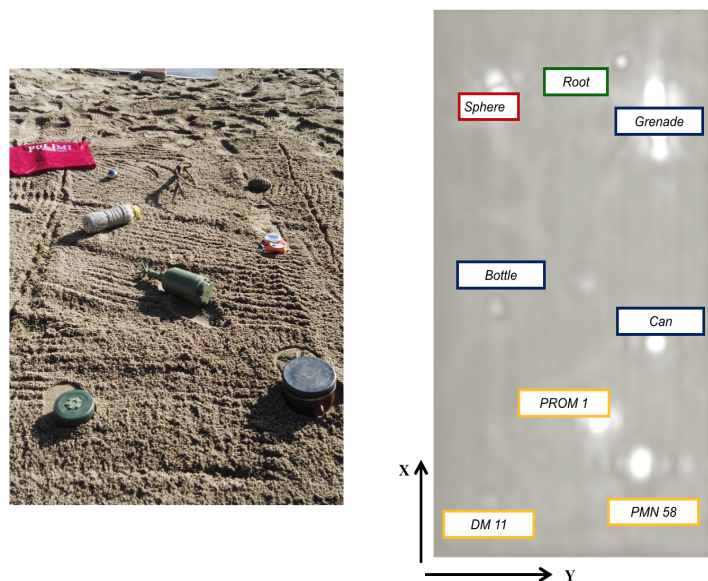


Figure 5.4: Acquisition setup G_2 .

5.1.2 Giuriati

The second dataset, called Giuriati and indicated with S_2 , was built utilizing the long-jump pit within the Giuriati sports campus as the test site in June 2017 (i.e., a minimal content of water in the soil mixture). The rationale behind this choice is twofold: first, the simplicity of bury and dig up objects; moreover, it reflects a typical situation of contaminated areas over the world. The soil was characterized by shallow clay content and a gritty texture. Even if, this setup seems to resemble the previous one, the different wheater conditions during the acquisitions made the mixture of the two subsoils really different in terms of content of water. Since the water has a relative permittivity of 81, while for the sand is about 3, the permittivity of water-bearing soil (and thus the resulting images) is strongly influenced by its water content [27].

We acquired a set of images with the same system as the previous setup. The site is shown in Figure 5.4, and consists of 8 targets representing inert landmine models and non-threat object buried in long jump landing pit sand, at a depth of approximately 10 cm. In this setup, we acquired 64 B-scans. The Figure 5.2 shows the characteristics of the simulated landmines

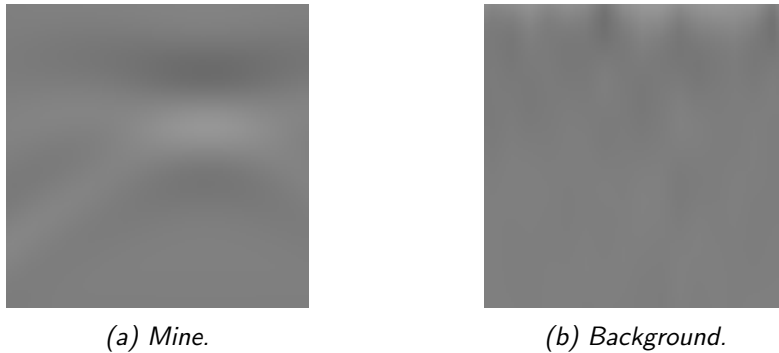


Figure 5.5: Examples of two synthetic patches.

the and clutter objects used in the experiments.

5.1.3 Synthetic data

The baseline used to compare our results and described later in this Chapter, make use of synthetic images together with real acquisitions. The set of synthetic images was constructed using a simulation software: `gprMax` [32]. The dataset contains a total number of 100 000 patches, either containing hyperbolic traces or just background noise. The images containing a landmine were labeled as $l = 1$, while the images containing only background noise were labeled as $l = 0$. Figure 5.5 show an example of two synthetic images: Figure 5.5a contains a hyperbola whether Figure 5.5b show the background noise. The simulation software allows simulating different real case scenario tuning some parameters. It is possible to specify both the instrumental setup (i.e., the radar's antennas, the time and spatial sampling frequency, etc.) and the physical properties of the buried objects as well as the soil (e.g., the dielectric and magnetic permittivity or the electric conductivity, the water content of the soil, etc.). The choices of this parameters were made to mimic as close as possible the setup used during the GPR acquisitions, trying to produce different cases in terms of the number of buried objects and soil properties.

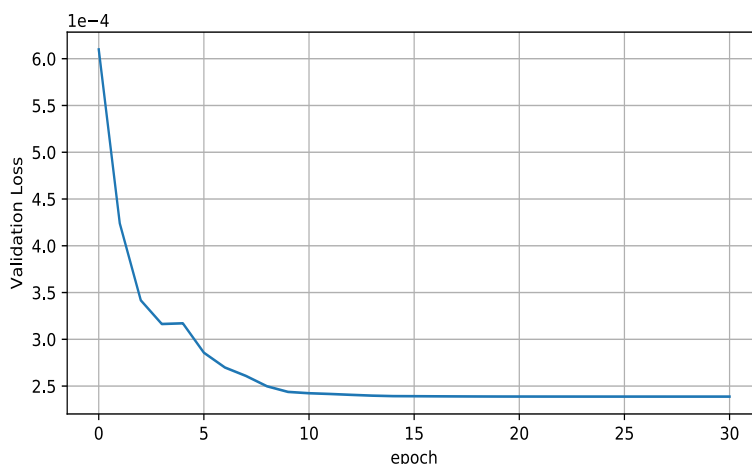


Figure 5.6: Validation Loss vs Epochs.

5.2 Experimental setups

With the objective of validating the performance of the pipeline described in Section 4, we perform a set of experiments testing the three different architectures outlined in Section 4.4. We divided our investigation into two groups of experiments: we first investigate the system trained and tested on data coming from the same dataset. Finally, we look at the results obtaining when the test was performed on data coming from a different data.

For each carried out experiment, we split the original dataset in \mathcal{D}_{val} , \mathcal{D}_{train} and \mathcal{D}_{test} . Specifically, the patches extracted from five B-scans went to populate \mathcal{D}_{val} and \mathcal{D}_{train} , 50% for validation and 50% for train, the remaining were used for the test phase.

Took an image from \mathcal{D}_{train} , we perform a patch extraction to obtain N patches for each original B-scan. We performed a minimal preprocessing. In particular, all the patches were normalized within the range $[-1, 1]$. Removal of the direct path. All the experiments used Adam optimizer with Back Propagation algorithm for training. The training was performed for 100 epochs each training, stopped if no improvement in the validation loss is achieved for 10 epochs. Figure 5.6 show an exemple of the Validation Loss evolution refer to the epochs. All networks have been implemented using Keras framework [11] backend with TensorFlow library [1]. All tests were run on a workstation equipped with a Titan X GPU reaching convergence

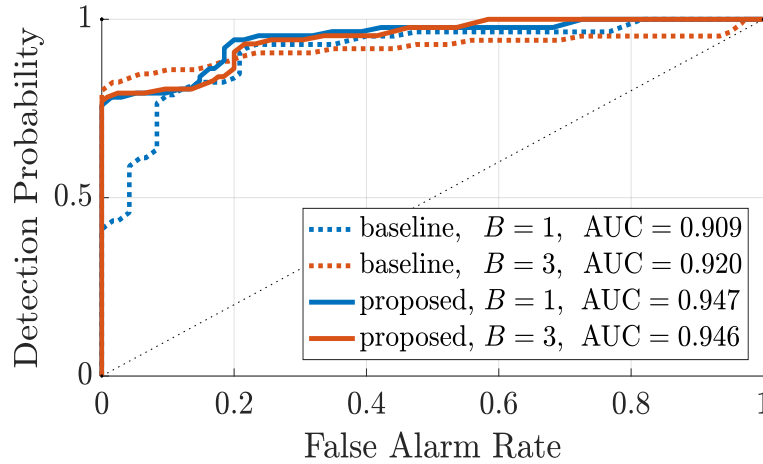


Figure 5.7: ROC obtained for different training set size B in comparison to the baseline

in a few minutes.

5.2.1 Evaluation methodology

The proposed method is based on a threshold Γ (4.1). Therefore, we evaluated our method employing receiver operating characteristic (ROC) curves.

The ROC curves represent detection probability and false alarm rate for different values of Γ . Detection probability represents the percentage of B-scans containing an object correctly detected as such. False alarm rate represents the probability of detecting an object into a B-scan that does not contain it. ROCs whose area under the curve (AUC) tends to 1 characterizes good detectors. A random guess is characterized by AUC equal to 0,5. As additional metrics, we also provide detector accuracy for the best selected Γ .

5.2.2 Baseline

With the objective of having a challenging baseline as a starting point to compare our results, we chose the result obtained in [17]. These experiments were carried out employing synthetic data (i.e., images generated with a simulation software) and real data, the latter were all taken from the Cranfield test site. For the training dataset, were used 50 000 background patches and 50 000 patches containing hyperbola from the synthetic dataset, to-

gether with a set of patches extracted from N real images lacking hyperbola traces from the real dataset. Specifically, we build three training datasets of images taken from Cranfield varying the number of real B-scans mixed with the synthetic ones from 1 up to 5. The test dataset otherwise was composed only by the unused real-data from the same test site. This method also used receiver operating characteristic (ROC) curves as the evaluation metric. This baseline with the best parameters choices reaches 90 % of accuracy and an ACU equal to 0.92. All the results were compared to this baseline.

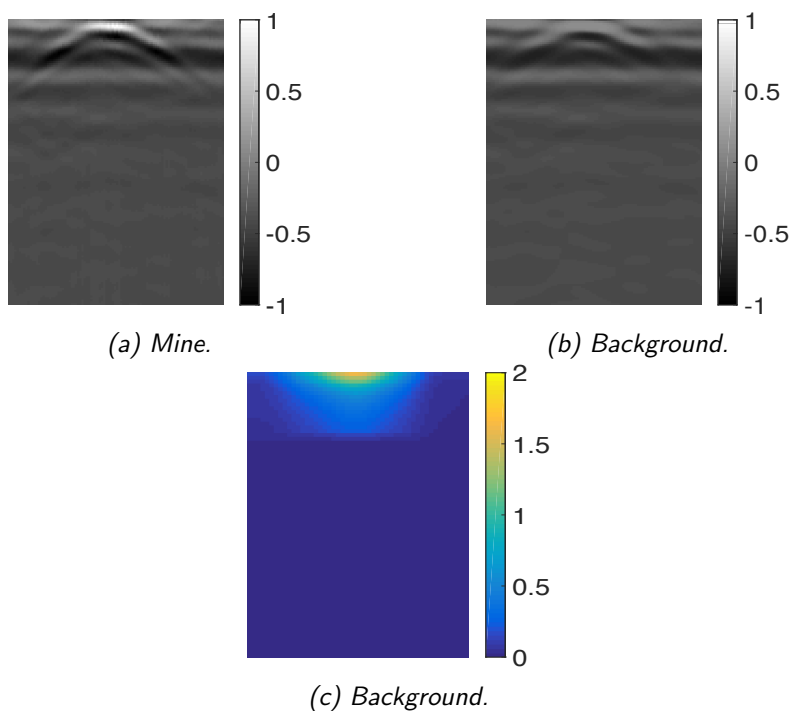


Figure 5.8: Examples of two GPR acquisitions.

5.3 Numerical Results

To provide a visual example of the working method, Figure 5.8 shows a B-scan region \mathbf{X} , its encoded and decoded version $\hat{\mathbf{X}}_j$, as well as the patch-by-patch error e_n obtained using architecture \mathcal{A}_1 with patches of size $B = 32$, (i.e., 32×32). It is possible to notice that the original hyperbola due to a buried object is just only mildly reconstructed in $\hat{\mathbf{X}}_j$. Conversely, the rest of the B-scan is almost perfectly autoencoder. Computing e_n it is possible to clearly spot an area with high mean square error, corresponding to the detected hyperbola.

5.3.1 Single dataset approach

The algorithm we choose as the baseline has a dependence, in terms of accuracy, from the number of real B-scans used in training. Therefore, we choose to investigate the effect of using more training data. We tested our architecture \mathcal{A}_1 using 1 and 3 training B-scans. Figure 5.7 shows the ROC

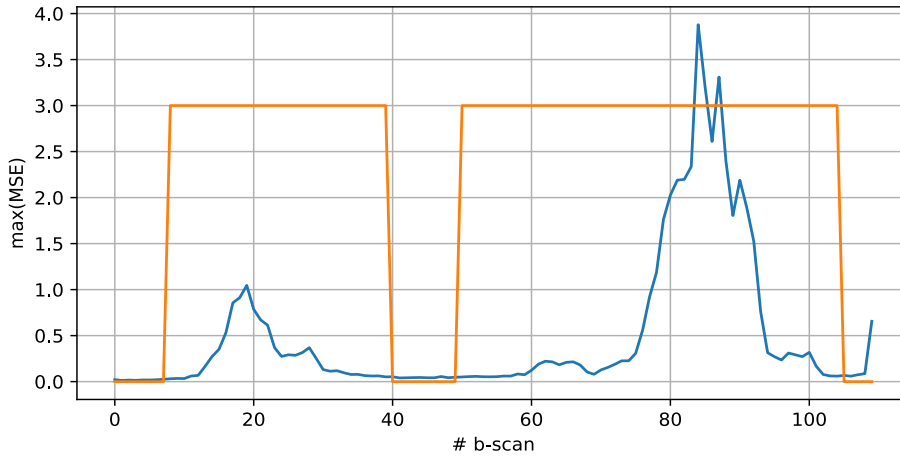


Figure 5.9: MSE vs ground truth

curves for our method and the baseline in the same exact conditions. It is possible to notice that our method improves over the baseline when only a few data are available for training (notice that the baseline used about 100 000 synthetic patches together with the B-scans). Therefore, we apparently do not need to use a high number of training images, as results using 1 or 3 B-scans are comparable. It is also worth noting that the baseline 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{A}_1 and changing the image patch size considering $B \in \{16, 32, 64\}$ and the stride in $\{4, 16, 32\}$. Table 5.1 reports the numeric results in terms of accuracy and AUC. It can be observed as the results remain in line with those presented in Figure 5.7 with a maximum AUC deviation of 1%. We, therefore, stopped our investigation on patch size, considering $B = 32$ a good choice. For this experiments we report in Figure 5.9 the MSE obtained for each analyzed B-scans compared with the ground truth (i.e., the detection human made). It can be notice how, if we imaginary move over our test site, from the starting line (5.1) following the GPR profile, crossing an area above a landmine results in an increase of the MSE.

Moreover, we tested the different architectures \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 on setup S1 using $B = 3$ training images. Also, in this case, we obtained comparable results, with slight AUC decrease for \mathcal{A}_2 , which reduces data dimensionality

too much. For this reason, we decided to only consider \mathcal{A}_1 for other tests.

Table 5.1: Impact of different patch parameters on architecture \mathcal{A}_1 , trained with $B = 5$ B-scans of \mathcal{S}_1 dataset.

Parameters		AUC
patch size = 32	stride 4	0.9470
	stride 16	0.9455
	stride 32	0.9285
patch size = 64	stride 4	0.9360
	stride 16	0.9320
	stride 32	0.9130
patch size = 128	stride 16	0.8646

Table 5.2: Impact of the different numbers of B-scans for training on dataset of \mathcal{S}_1 , patch size 64×64 , and stride 4×4 .

Training b-scans	\mathcal{A}_1	\mathcal{A}_2	\mathcal{A}_3
$N = 1$	0.9344	0.9247	0.9468
$N = 3$	0.9329	0.9195	0.8956
$N = 5$	0.9360	0.9230	0.9320

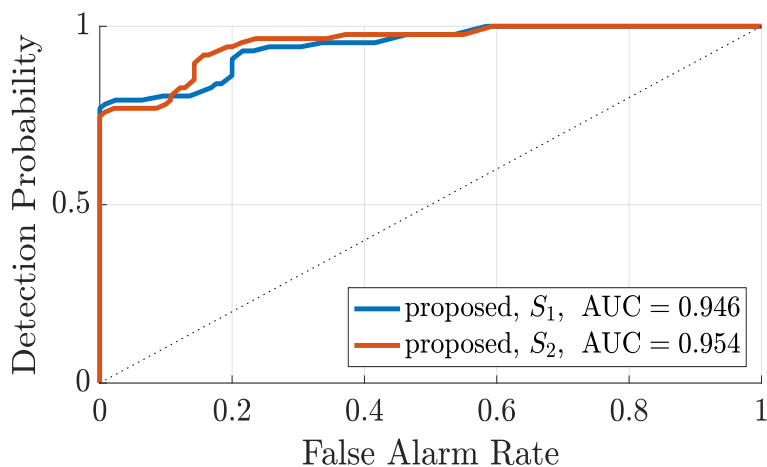


Figure 5.10: Proposed solution trained on setups S_1 and S_2 , then tested on S_1 .

5.3.2 Cross-dataset approach

The results achieved in the previous experiments demonstrate the efficacy of our model. Until now, we make use of real data with a significant constraint: all the acquisitions were taken on the same day at the same test site (i.e., the dielectric properties of the soil was the same). In a comprehensive study, it whort to investigate how the system reacts when is tested with different acquisitions (i.e., captured a different day or on a different test site). Therefore, as a conclusion of our investigation, we performed a cross-dataset test. Specifically, that involve using different data during the training and testing phase. The scope of this experiments was to prove that the model can be training-site independent. We trained \mathcal{A}_1 on $B = 3$ B-scans split into 32×32 patches using either setup S_1 or S_2 , then we tested on on S_1 . Figure 5.10 provides some better insights into the role of using different datasets on landmine detection. From this figure and numerical results reported in Table 5.3, 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.

Table 5.3: Best results of cross-tests between \mathcal{S}_1 and \mathcal{S}_2 .

		TEST	
		\mathcal{S}_1	\mathcal{S}_2
TRAIN	\mathcal{S}_1	0.9755	0.9321
	\mathcal{S}_2	0.9610	0.9391

5.3.3 Results comments

The result shows an improvement of the state-of-the-art in terms of a better accuracy in discriminating a landmine from its background. A confront with the baseline show how the performance (i.e., the AUC) increase of 0.04, even if our system does not consider information about landmines. This is really important since it is almost impossible to capture B-scans above a landmine in a real minefield. Moreover, we use just five acquisitions to train our system, thus avoiding the cost of acquiring a big amount of data. The cross-dataset tests reveal that even in the case of absence of background data of the field under analysis, our system can be trained on different soil and perform well on the new minefield.

Chapter 6

Conclusions

A complete landmines detection system is composed of the following modules: detection, recognition and localization. In this thesis, we proposed an anomaly detection technique based on Convolutional Autoencoders for landmine detection in GPR data. Specifically, we focus on the task of detection (i.e., detect whether any target is buried within an area of interest) implementing a classification system that discriminates the images containing pure background from the ones depicting traces of objects. We implemented the autoencoder architectures using Keras, a library to build and train neural networks in TensorFlow. Specifically, we built various architectures able to shrink the image dimension by a factor, thus capturing only the salient feature of the training data. Then, we performed the training on B-scans containing only the information relative to the soil properties and performed some test with unseen images. The proposed solution is a completely data-driven approach exploiting a one-class paradigm. As a matter of fact, our system uses only data not containing landmine traces at the training stage. With our approach, it is possible to acquire some B-scans of controlled mine-free fields, and deploy the system to detect never seen before objects. This makes the system robust to a wide variety of targets, as no strong a priori assumptions are needed. 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.

Moved by the last evidence, we considered the possibility of training the system with data coming from a particular soil and test it on acquisitions with different properties. We see that our system generalizes well on data outside the training dataset, proving that our system was not affected by overfitting (i.e., the excellent performance were not a mere consequence of the specific train dataset). This implies that for a practical application of our system will be possible to pre-train the autoencoder on a controlled site with no treats and then using the same model for detection on different geographical areas. The results with a cross-dataset approach reach performance close to the one-class, with a slightly drop of just 0.02 % in accuracy.

6.1 Future developments

Future developments include the addition of the demining system tasks not investigated in our work, such as localization (i.e., determine the precise location of targets of interest) and discrimination (i.e., discriminate whether at least one of the detected object is a landmine).

In our work, we focus only on the direct polarization data of the GPR. Nonetheless, the GPR allows capturing data with different polarizations while performing a single acquisition; the produced output is, therefore, a set of data containing the signals from the direct polarizations (i.e., HH and VV) together with the cross polarization HV, VH. The information coming from the different polarizations can be different in the case of a buried object with no symmetrical geometry (i.e., a wooden stick or a pipe). Therefore, including this consideration into the design can facilitate the discrimination of landmines from other clutters. A future improvement can consider all the polarization to improve the classification task.

We show the advantages of a system tailored to process single B-scan at a time. A different approach can consider working with 3D volumes (i.e., the volume is a 3D representation of the subsoil obtained by flanking the B-scans together). Future works include investigating this alternative.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Rasaan Bello. Literature review on landmines and detection methods. *Frontiers in Science*, 3(1):27–42, 2013.
- [3] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.
- [4] Lance E. Besaw and Philip J. Stimac. Deep learning algorithms for detecting explosive hazards in ground penetrating radar data. -, 9072:90720Y, 2014.
- [5] Lance E Besaw and Philip J Stimac. Deep convolutional neural networks for classifying gpr b-scans. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, volume 9454, page 945413. International Society for Optics and Photonics, 2015.

-
- [6] Giovanni Borgioli, Lorenzo Capineri, PierLuigi Falorni, Serena Matucci, and Colin G Windsor. The detection of buried pipes from time-of-flight radar data. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2254–2266, 2008.
- [7] Dragana Carevic. Clutter reduction and target detection in ground-penetrating radar data using wavelets. In *Detection and Remediation Technologies for Mines and Minelike Targets IV*, volume 3710, pages 973–979. International Society for Optics and Photonics, 1999.
- [8] Dragana Carevic. Clutter reduction and detection of minelike objects in ground penetrating radar data using wavelets. *Subsurface Sensing Technologies and Applications*, 1(1):101–118, 2000.
- [9] Nigel J Cassidy and HM Jol. *Electrical and magnetic properties of rocks, soils and fluids*, volume 2. chapter, 2009.
- [10] Huanhuan Chen and Anthony G Cohn. Probabilistic robust hyperbola mixture model for interpreting ground penetrating radar data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010.
- [11] François Chollet et al. Keras. <https://keras.io>, 2015.
- [12] Davide Cozzolino and Luisa Verdoliva. Single-image splicing localization through autoencoder-based anomaly detection. *8th IEEE International Workshop on Information Forensics and Security, WIFS 2016*, 2017.
- [13] David J Daniels. *Ground penetrating radar*, volume 1. Iet, 2004.
- [14] David J. Daniels. Ground penetrating radar for buried landmine and ied detection. In *Unexploded Ordnance Detection and Mitigation*, pages 89–111. Springer, 2009.
- [15] Hichem Frigui and Paul 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*, 17(1):185–199, 2009.

-
- [16] Umar Shahbaz Khan and Waleed Al-Nuaimy. Background removal from gpr data using eigenvalues. In *Ground Penetrating Radar (GPR), 2010 13th International Conference on*, pages 1–5. IEEE, 2010.
- [17] Silvia Lameri, Federico Lombardi, Paolo Bestagini, Maurizio Lualdi, and Stefano Tubaro. Landmine Detection from GPR Data Using Convolutional Neural Networks. *25th European Signal Processing Conference (EUSIPCO) Landmine*, 2017.
- [18] International Campaign To Ban Landmines. Landmine monitor 2016. Technical report, Human Right Watch, 2016.
- [19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [20] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155, 1989.
- [21] Yuwei Liao, Loren W. Nolte, and Leslie M. Collins. Decision fusion of ground-penetrating radar and metal detector algorithms—a robust approach. *IEEE Transactions on Geoscience and Remote Sensing*, 45(2):398–409, 2007.
- [22] J. Daniels L.P. Peters and J. Young. Ground Penetrating Radar as a Subsurface Environmental Sensing Tool. *Proceedings of the Ieee.*, 82(2):1802–1822, 1994.
- [23] Maurizio Lualdi. 3d acquisition using gpr over small areas: A cost effective solution. In *24rd EEGS Symposium on the Application of Geophysics to Engineering and Environmental Problems*, 2011.
- [24] T. Mitchell. *Machine Learning.*, volume 45. McGraw Hill., 1997.
- [25] Francesco Picetti, Giuseppe Testa, Federico Lombardi, Paolo Bestagini, Maurizio Lualdi, and Stefano Tubaro. Convolutional Autoencoder for Landmine Detection on GPR Scans. In *41st IEEE Conference on Telecommunications and Signal Processing (TSP) cicco*, 2018.

- [26] © 2018 Oxford University Press. Definition of "machine learning" at oxforddictionaries.com. https://en.oxforddictionaries.com/definition/us/machine_learning.
- [27] DA Robinson, Scott B Jones, JM Wraith, Daniel Or, and SP Friedman. A review of advances in dielectric and electrical conductivity measurement in soils using time domain reflectometry. *Vadose Zone Journal*, 2(4):444–475, 2003.
- [28] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [29] Peter M Roth and Martin Winter. Survey of appearance-based methods for object recognition. *Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Technical Report ICGTR0108 (ICG-TR-01/08)*, 2008.
- [30] DE Rumelhart, GE Hinton, and RJ Williams. Neurocomputing: Foundations of research. chap. learning representations by back-propagating errors, 696–699, 1988.
- [31] Peter A Torrione, Kenneth D Morton, Rayn Sakaguchi, and Leslie M Collins. Histograms of oriented gradients for landmine detection in ground-penetrating radar data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1539–1550, 2014.
- [32] Craig Warren, Antonios Giannopoulos, and Iraklis Giannakis. gprMax: Open source software to simulate electromagnetic wave propagation for Ground Penetrating Radar. *Computer Physics Communications*, 209:163–170, 2016.
- [33] Thomas R. Witten. Present state of the art in ground-penetrating radars for mine detection. In *Detection and Remediation Technologies for Mines and Minelike Targets III*, volume 3392, pages 576–586. International Society for Optics and Photonics, 1998.

Appendix A

Submission to TSP for GPR Applications

We include the paper that summarizes our results and we that submitted to the 41st IEEE Conference on Telecommunications and Signal Processing (TSP), held in Athens, Greece, 4/6 July 2018.

Convolutional Autoencoder for Landmine Detection on GPR Scans

Francesco Picetti¹, Giuseppe Testa¹, Federico Lombardi², Paolo Bestagini¹, Maurizio Lualdi³, Stefano Tubaro¹

¹Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano - Milano, Italy

²Department of Electrical and Electronic Engineering, University College London - London, United Kingdom

³Dipartimento 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.

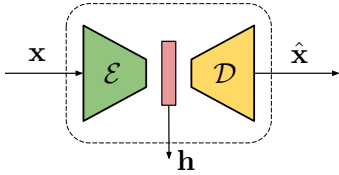


Fig. 1: Scheme of an undercomplete autoencoder. The encoder \mathcal{E} turns the input \mathbf{x} into its hidden representation \mathbf{h} . This is turned into $\hat{\mathbf{x}}$ by the decoder \mathcal{D} .

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 $\hat{l} \neq l$.

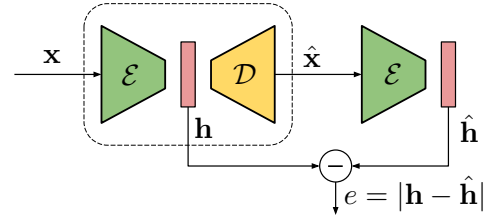


Fig. 2: Diagram of the proposed anomaly detection scheme. A patch under analysis \mathbf{x} is autoencoded to $\hat{\mathbf{x}}$ and encoded again into $\hat{\mathbf{h}}$. Anomaly is detected by thresholding e value.

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.

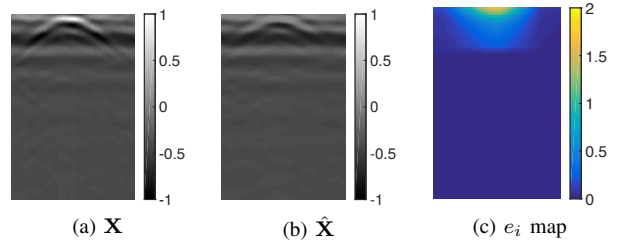


Fig. 3: Example of an original B-scan \mathbf{X} showing an hyperbola (a), its autoencoded version $\hat{\mathbf{X}}$ in which the hyperbola is not perfectly reconstructed (b), and spatial reshape of e_i related to B-scan patches.

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.

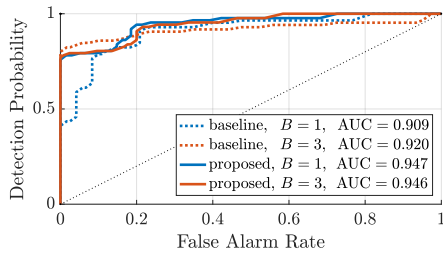
As explained during numerical analysis, we constructed different training datasets by changing the amount of considered training B-scans and setups. For testing, we always considered all B-scans non used for training belonging to setup S_1 only.

V. EXPERIMENTAL RESULTS

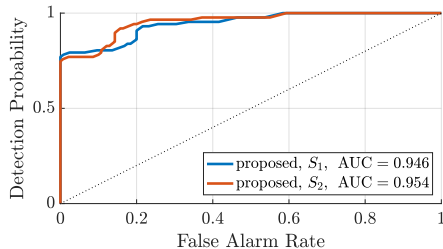
In this section, we explain the used evaluation metrics and collect results from our numerical analysis.

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 32×32 . 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 32×32 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 16×16 , 32×32 and 64×64 . 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 32×32 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 32×32 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.