Online Dictionary Learning for Classification of Antipersonnel Landmines using Ground Penetrating Radar

DISSERTATION

zur Erlangung des Grades eines Doktors

der Ingenieurwissenschaften

vorgelegt von:

Dipl.-Ing. Fabio Giovanneschi

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät

der Universität Siegen

Siegen 2020

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

Betreuer und erster Gutachter:

Prof. Dr.-Ing. Joachim Ender

Universität Siegen

Zweiter Gutachter:

Prof. Dr.-Ing. Otmar Loffeld

Universität Siegen

Tag der mündlichen Prüfung:

9 Juli 2020

To Marian, my parents and Lucas.

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Joachim Ender for the support of my Ph.D study and related research. Prof. Ender made very important contributions to the Radar community and he has a very vast knowledge in Radar signal processing; I am grateful for having him as advisor and mentor for my Ph.D study. He followed my career in Fraunhofer FHR from the beginning, we worked together in different projects and I had the chance of knowing him personally. Being around him has been always motivating and inspiring and we also shared a good amount of stories and laughs!

I would like to thank Prof. Loffeld, for his insightful comments and encouragement. He welcomed me on the IPP doctoral progam in the ZESS department in Siegen demonstrating genuine interest in my PhD work, for that I am grateful. His questions and comments motivated me to sharpen part of my research and helped in improving the final PhD presentation.

My thanks also go to Dr. Udo Uschkerat. Without his support in the first half of my PhD journey, it would not have been possible to conduct this research. He provided me the necessary resources, he shared his knowledge in the Ground Penetrating Radar field and partecipated in the initial part of the work contained in this thesis.

I thank Stefan Brüggenwirth for his support of my PhD work from 2015 on. He allowed me the time to conduct my research while leading the Cognitive Radar department during its first challenging times until now. He always demonstrated interest in my work and helped me in promoting it.

I am honoured to have worked together with Prof. Yonina Eldar. Her contribution to the Radar community is outstanding, expecially on the topic of Compressive Sensing. Our cooperation resulted in the pubblication of the articles which consituted the key contribution of this PhD thesis. Her challenging comments and insights greatly improved the quality of this work.

A very special thank goes to Kumar Vijay Mishra who was working with Prof. Eldar during the development of this work and joined me side by side in developing the Online Dictionary Learning strategy for Antipersonnel Landmine Recognition with GPR . We met ourselves in Israel under a very challenging situation which turned out to be the start of a very fruitful collaboration and friendship. I thank him for the stimulating discussions, the sleepless nights we were working together before deadlines and for all the anedoctes we shared in the last years.

All of this would not have been possible without Maria Antonia Gonzalez Huici. Not only she contributed to the work contained in this Thesis, Marian also helped me trough all the difficult efforts which came during this PhD path. There have been many times where disconfort and boredom took the best of me during these years and she was always there cheering me up and putting things in the right perspective. I thank her for all the love she gave and is giving to me, I feel extremely lucky for sharing my life with her and our beautiful son Lucas. Te quiero mucho Marian.

I want to deeply thank all the friends I've made during this journey, some of them left some of them stayed; in general they contributed to made my life outside the shores of Livorno pleasant and rich. Grazie!

And last but not the least, I would like to thank my parents. I thank them for letting me choose my path freely and supporting me throughout all my life, including this PhD journey.

Zusammenfassung

Die Kontamination von Antipersonenlandminen (APM) auf der ganzen Welt ist seit vielen Jahrzehnten ein ernstes Problem für die Menschheit. Ab 1999, als die Landmine Monitor-Organisation gegründet wurde, gab es mehr als 110.000 registrierte Opfer, während 2018 einer der höchsten Prozentsätze der betroffenen Zivilisten war (87%, 47% davon Kinder). APMs sind relativ klein (z. B. ca. 10 cm Durchmesser und 5 cm Höhe) und werden im Allgemeinen durch leichten Druck aktiviert, sodass sie in unmittelbarer Nähe töten (oder verstümmeln). Da APMs fast ausschließlich aus Kunststoff bestehen (mit Ausnahme der Zündvorrichtung und anderer Kleinteile), können sie mit Metalldetektoren häufig nicht erkannt werden.

Eine vielversprechende Alternative ist der Einsatz des Ground Penetrating Radar (GPR). GPRs sind für den dielektrischen Kontrast empfindlich, daher können sie nichtmetallische vergrabene Ziele nur aufgrund ihrer unterschiedlichen dielektrischen Eigenschaften vom Hintergrundboden unterscheiden. Um die erforderliche Auflösung für die identifizierungvon APMs zu erhalten, muss ein GPR für die Erkennung von Landminen mit sehr hohen Frequenzen und Bandbreiten (beide bei 1 bis 2 GHz) arbeiten. In diesen Situationen gibt es viele unerwünschte Rückstreuungsbeiträge, die die tatsächliche Zielantwort maskieren können.

Klassifizierungsansätze werden oft verwendet, um zwischen Landminen und natürlichen oder künstlichen Störquellen zu unterscheiden. Zu diesem Zweck wird normalerweise eine Datenbank mit Zielsignaturen von bekannten APMs verwendet, um den Klassifikator der Wahl zu trainieren. Klassifizierungstechniken erfordern jedoch in der Regel eine Sammlung von Daten, die alle möglichen Ziele und Bodeneigenschaften darstellen. Eine solche allgemeine Datenbank zu erstellen, bleibt eine mühsame Aufgabe. Es ist häufig vorzuziehen, Unterscheidungsmerkmale (die gegenüber der Variation der Szenariobedingungen robust bleiben) aus einer kleineren (aber repräsentativen) Datenbank zu extrahieren. Die für diese Arbeit ausgewählten Merkmale sind spärliche Repräsentationskoeffizienten.

Sparse Representation (SR) teilt seinen technischen Rahmen mit der Compressive Sensing Theorie (CS). Während das Ziel von CS darin besteht, das interessierende Signal unter Verwendung weniger seiner Abtastwerte wiederherzustellen, befasst sich SR mit dem Problem der Darstellung eines Signals mit einer minimalen Anzahl von Koeffizienten in einer bestimmten Basis oder einem bestimmten Dictionary. Diese wenigen Koeffizienten können für eine bestimmte Zielklasse charakteristisch sein (in unserem Fall verschiedene Arten von Minen oder Clutter), wenn das ausgewählte Wörterbuch Beispiele für alle Klassen im betrachteten Datensatz enthält.

Die Auswahl des Wörterbuchs ist entscheidend; Anstatt es auszuwählen, kann man es aus einem repräsentativen Satz von Signalen lernen, nämlich einem training set. Dictionary Learning (DL) Technikenzielen darauf ab, ein dictionary zu generieren, das verschiedene Klassen von Zielen mit einer minimalen Anzahl von non-zero-Koeffizienten darstellen kann. Beliebte DL-Techniken (wie K-SVD) behandeln den gesamten Trainingssatz in jeder Iteration, wodurch der Lernprozess für hochdimensionale Datensätze verlangsamt wird. Dieser Ansatz wird als Batch Dictionary Learning bezeichnet. Online-DL-Techniken bewältigen den Trainingssatz, indem sie ein Element nach dem anderen oder in Mini-batches betrachten. Dadurch wird die Lernverarbeitung schneller als bei Batch-DL und an Variationen anpassbar.

In dieser Arbeit analysieren wir eine Auswahl von state-of-the-art Online-DL-Ansätzen zur spärlichen darstellungsbasierten Klassifizierung von verschütteten Landminen unter Verwendung von von GPR-range profiles (oder A-Scans). Eine detaillierte Beschreibung der Entwicklung eines neuartigen Online-Dl-Algorithmus, des Drop-Off-Mini-Batch-Online-Wörterbuchlernens (DOMINODL), wird ebenfalls vorgestellt. Für die Validierung verwenden wir range profiles aus einem experimentellen GPR-Datensatz, der verschiedene Klassen von Landminen-Simulanzien enthält, die in einem sandigen und stark inhomogenen Boden vergraben sind.

Die vorgeschlagene Strategie besteht zunächst darin, eine Sammlung sorgfältig ausgewählter Bereichsprofile als Trainingssatz für verschiedene Online und Batch DL Strategien (K-SVD, LRSDL, ODL, CBWLSU und DOMINODL) zu verwenden. DL-Algorithmen reagieren sehr empfindlich auf Eingabeparameter (z. B. Anzahl der Iterationen, Anzahl der gelernten Atome usw.). Um eine optimale Auswahl der Parameterkombination zu gewährleisten, haben wir eine umfassende Auswertung durchgeführt, die sich auf statistische Kennzahlen wie die Kolmogorov-Smirnoff-Test distance und die Dvoretzky-Kiefer-WolfowitzInequality stützt. Die mit den gelernten Wörterbüchern erhaltenen spärlichen Koeffizienten werden schließlich als Merkmale für einen Support Vector Machine Classifier verwendet.

Die Online-DL-Strategie erwies sich als viel schneller als die ihrer Batch-Kollegen. Insbesondere ist DOMINODL das schnellste und kann das Wörterbuch in nur 1,75 Sekunden erlernen (3-mal schneller als ODL und 15-mal schneller als K-SVD). Bei Verwendung der optimalen Eingabeparameter für DL stellen wir fest, dass größere Minen (PMN / PMA2) mit ausgezeichneter Genauigkeit klassifiziert werden ($P_{cc} = 98\%$), und dies gilt auch für Clutter ($P_{cc} = 90)\%$). Mittelgroße Minen (ERA) sind nicht immer korrekt klassifiziert, aber die Erklärungen um ihre Grundwahrheit entsprechen immer einer Klasse von Landminen. Die Klassifizierungsgenauigkeit für kleinere Landminen (Typ 72) ist in Bezug auf die EFR-Ziele höher, obwohl es sich um die kleinsten Minen im Satz handelt. Online-DL-Methoden zeigen höhere Werte für P_{CC} für die ERA- und Type-72-Ziele in Bezug auf K-SVD und LRSDL. Als zusätzlichen Vergleich mit einem Klassifizierungsalgorithmus nach dem neuesten Stand der Technik verwenden wir ein Convolutional Neural Network (CNN), das mit demselben Trainingssatz trainiert wurde, der für unseren DL-basierten Ansatz verwendet wurde. Klassifizierungsergebnisse mit CNN sind im Allgemeinen schlechter, insbesondere für ERA und T72 (P_{cc} verschlechtert sich um mehr als 27%). Unsere abschließende Bewertung bestand in der Beurteilung der Klassifizierungsgenauigkeit, wenn die ursprünglichen Stichproben der GPR-Bereichsprofile auf 25% 50% und 75% reduziert wurden. Während CNN seine Klassifizierungsgenauigkeit drastisch reduziert, selbst wenn die Reduktion nur 25% beträgt, sind DL-basierte Ansätze gegenüber der Probenreduktion definitiv robuster. Der vorgeschlagene Ansatz ist nicht auf den GPR-Datensatz beschränkt und kann auch auf anderen Radaranwendungen getestet werden. Aufgrund der extrem schnellen Lernzeit von Online-DL-Strategien (insbesondere DOMINODL) ist auch ein Echtzeitlernen eines sich ständig ändernden Trainingssatzes (Aktualisierung mit neuen Messungen) vorgesehen, was den Weg für einen vollständig kognitiven Klassifizierungsansatz ebnet.

Abstract

The contamination of Antipersonnel Landmines (APM) all over the world has been a serious threat to mankind since many decades. Starting from 1999, when the Landmine Monitor organization was founded, there have been more that 110,000 registered casualties while 2018 marked one of the highest percentage of civilians affected (being 87% with 47% of them being children). APMs are relatively small (for instance, ca. 10cm of diameter and 5cm of height), generally activated by light pressure hence designed to kill (or maim) in their close proximity. Being many APMs almost entirely made of plastic (with the exception from their ignition device and other small parts) they often cannot be recognized using metal detectors.

A promising alternative consists in the use of the Ground Penetrating Radar (GPR). GPRs are sensible to dielectric contrast, hence they can distinguish non-metallic buried targets from the background soil only due to their different dielectric properties. To obtain the necessary resolution to resolve APMs, a GPR for landmine recognition has to work with high frequencies and bandwidth (being both around 1-2 GHz). In these situations there are many unwanted backscattered contributions that can mask the actual target response.

Classification approaches are often employed to recognize between landmines and natural or man-made clutter sources; in order to do that, a database of target signatures from known APMs is usually employed to train the classifier of choice. However, classification techniques typically require a collection of data representing all possible targets and soil characteristics; to create such a general database remains a cumbersome task. It is often preferable to extract discriminative features (which will remain robust to the variation of the scenario conditions) from a smaller (but representative) database; the features chosen for this work are sparse representation coefficients.

Sparse Representation (SR) shares its framework of techniques with Compressive Sensing theory (CS). However, while CS goal is to recover the signal of interest by using less of its samples, SR deals with the problem of represent a signal with a minimum number of coefficients in a certain base or dictionary. These few coefficients can be characteristic of a certain class of target (in our case different types of mines or clutter) if the chosen dictionary contains sufficient examples of all classes in the considered dataset.

The selection of the dictionary is crucial; instead of choosing it, one can learn it from a representative set of signals, namely a training set. Dictionary Learning (DL) techniques aim to generate a dictionary which can represent different classes of targets with a minimum number of non-zero coefficients. Popular DL techniques (such as K-SVD) deal with the entire training set in each iteration, making the learning process slow for high dimensional datasets; this approach is called batch Dictionary Learning. Online-DL techniques deal with the training set by considering one element at a time or in mini batches, making the processing of learning faster than batch-DL and adaptive to variations.

In this work we anaylize a selection of state-of-the-art Online-DL approaches for sparse representation based classification of buried landmines using GPR range profiles (or A-Scans). A detailed description on the development of a novel Online-Dl algorithm, the Drop-Off Mini-Batch Online Dictionary Learning (DOMINODL), is also presented. The development of this algorithm is one of the main contribution of this thesis. For the validation, we use range profiles from an experimental

GPR dataset which includes different classes of landmine simulants, buried in a sandy and highly inhomogeneous soil.

The proposed strategy initially consists in using a collection of carefully selected range profiles as a training set for different Online and Batch DL strategies (K-SVD, LRSDL, ODL, CBWLSU and DOMINODL). DL algorithms are very sensitive to input parameters (such as number of iterations, number of learned atoms, etc.); in order to assure an optimal selection of parameters combination, we performed an exhaustive evaluation which relies on statistical metrics such as Kolmogorov-Smirnoff test distance and Dvoretzky-Kiefer-Wolfowitz inequality. The sparse coefficients obtained with the learned dictionaries are finally employed as features for a Support Vector Machine Classifier.

Online-DL strategy demonstrated to be much faster than their batch counterparts. In particular, DOMINODL is the fastest, being capable of learning the dictionary in just 1.75 seconds (3 times faster than ODL and 15 faster than K-SVD). When using optimal input parameters for DL, we observe that bigger mines (PMN and PMA2 types) are classified with excellent accuracy ($P_{cc} = 98\%$), and so it is for clutter ($P_{cc} = 90\%$). Medium sized mines (a standard test target provided by ERA technologies, we simply call them ERA) are not always correctly classified but the declarations around their ground truth always correspond to a class of landmine. Classification accuracy for smaller landmines (Type-72) is higher with respect to ERA targets, despite being the smallest mines in the set. Online DL methods show higher P_{CC} for the ERA and Type-72 targets with respect to K-SVD and LRSDL. As an additional comparison with a state-of-the art classification algorithm, we use a Convolutional Neural Network (CNN) trained with the same training set used for our DL-based approach. Classification results with CNN are poorer in general, especially for ERA and T72 (P_{cc} degrades by more than 27%). Our final evaluation consisted in assessing the classification accuracy when reducing the original samples of the GPR range profiles to 25% 50% and 75%. While CNN classification accuracy reduces drastically even when the reduction is only 25%, DL-based approaches are definitely more robust to the sample reduction.

The proposed approach is not limited to GPR dataset and can be tested on other radar applications as well. Due to the extremely fast learning time of Online-DL strategies (especially DOMINODL), real time learning of a constantly varying training set (updating it with new measurements) it is also envisioned, paving the way for a fully cognitive classification approach.

Contents

Appendices

Chapter 1

Introduction

The introduction is divided in two sections describing the main problem adressed in this work (de-mining via Ground Penetrating Radar) and the manifold of techniques which have been used to tackle it (the Compressive Sensing framework).

1.1 GPR and the de-mining problem

The international legal definition describe as *explosive remnants of war* (ERW) both unexploded Ordinances (UXO) and abandoned explosives ordnances (AXO). UXO are defined as unstable explosives which have been placed during conflicts and remained untriggered while AXO have not been used during armed conflicts but have been abandoned and are not effectively controlled. ERW can include artillery shells, grenades, mortars, rockets, air-dropped bombs and cluster munition remnants. ERW have a relatively big deflagration and most of them are anti-vehicle mines which are designed to explode from the presence of a vehicle.

Antipersonell landmines (APM) are not defined as ERW, though they represent a serious threat for many individuals (mostly civilians) and one of the worst kind of global pollution nowadays. As the name suggest, differently from ERW, APM are generally activated by a single person (wether a soldier or a civilian) and they are designed to kill (or maim) in their close proximity. APM are generally blast mines, fragmentation mines or even improvised explosive devices (IED), the latter are not constructed by conventional military designs and they are associated to insurgent guerrillas, commando forces, etc.

In November 2018, the landmine monitor (the reporting network which provides a global overview of the landmine situation) recorded 7239 mine/ERW casualties, of which at least 2793 people were killed (with 87% being civilian and 47% of them being children). Over the last years, APM casualties had a sharp increase. The year 2016 marked the highest number of annual recorded casualties in the monitor data since 1999 (9,228), the most child casualties ever recorded, and the highest number of annual casualties caused by improvised mines. Casualties in 2016 were identified in 52 states and four other areas, of which 35 are States Parties to the Mine Ban Treaty. The high total was mostly due to casualties recorded in armed conflicts in Afghanistan, Libya, Ukraine, and Yemen.

The Monitor has recorded more than 110,000 mine/ERW casualties since its global tracking began in 1999, including some 80,000 survivors (see figure 1.1) [1, 2].
The detection and efficient classification of APM remains nowadays a complex scientific and technical issue. The use of conventional metal-detectors in demining operations may become particularly slow and inefficient because of the low-metal content of several modern landmines and the presence of abundant metallic scrap in battlefields. Therefore, alternative methods have been intensively investigated in the past years [3]. Ultra wideband (UWB) Ground Penetrating Radar (GPR, hereafter) is a promising alternative and/or complementary technology to tackle this serious problem since it can sense any dielectric anomaly present in soil in a non-invasive way [4]. A GPR is used for probing the underground by transmitting radio waves in the subsurface and recording the backscattered reflections. The interest in GPR is due to its ability to reveal buried objects and detect non-metallic scatterers with increased sensitivity to dielectric contrast [5, 6]. This sensing technique is, therefore, attractive for several applications such as geophysics, archaeology, forensics, and defense (see e.g. [4, 5] for some surveys). Mine detection GPR usually operates in L-band (1-2 GHz) with ultra-wideband (UWB) transmit signals that allow resolving small targets (5-10 cm diameter) at shallow depths (15-30 cm) [7, 8]. In such situations, GPR target recognition suffers from signal distortion due to inhomogeneous soil clutter, surface roughness and antenna ringing. Moreover, the constituting material of many models of landmines is largely plastic and has a very weak response to radar signals due to its low dielectric contrast with respect to the soil, therefore, complex processing may become necessary to extract and interpret the target responses from the radar data [4, 9]. A variety of signal processing algorithms have been proposed for detection of low metal-content landmines in realistic scenarios; approaches based on feature extraction and classification are found to be the most effective (see e.g [10–13]), yet false-alarm rates remain very high. Further, a high-resolution GPR has long scan times thereby making the data acquisition by a portable instrument very cumbersome [14].

1.2 The Compressive Sensing framework

Sensing and processing systems for images, audio, video and other kind of data, rely on the sampling and digitalization of continuous, band-limited signals. The theoretical background provided by Kotelnikov(1933), Nyquist (1949) and Shannon (1949) indicated that such signals can be exactly recovered from a set of uniformly spaced samples taken with a rate which is twice the cutoff of the signal itself. The digital revolution began, and, since their first introduction, digital systems demonstrated to be cheaper, more flexible and robust respect to its analog counterparts. However, if we just think about radar applications, the demand for a large bandwidth results in very high Nyquist rates, increasing (when possible) the complexity and cost of A/D converters along with data dimensionality. Moreover, many radar systems rely on multiple acquisition channels thus increasing the amount of data by another factor. Despite the advances in computational power, the acquisition and processing of such an enormous amount of information is still very challenging [15].
Compressive Sensing (CS hereafter) is a novel signal processing framework which has rapidly evolved in the recent years. Its development began with the necessity to overcome the limitations of the classical sampling theory.
To better understand CS, we need to address the concept of *sparsity*. We say that a signal is sparse in

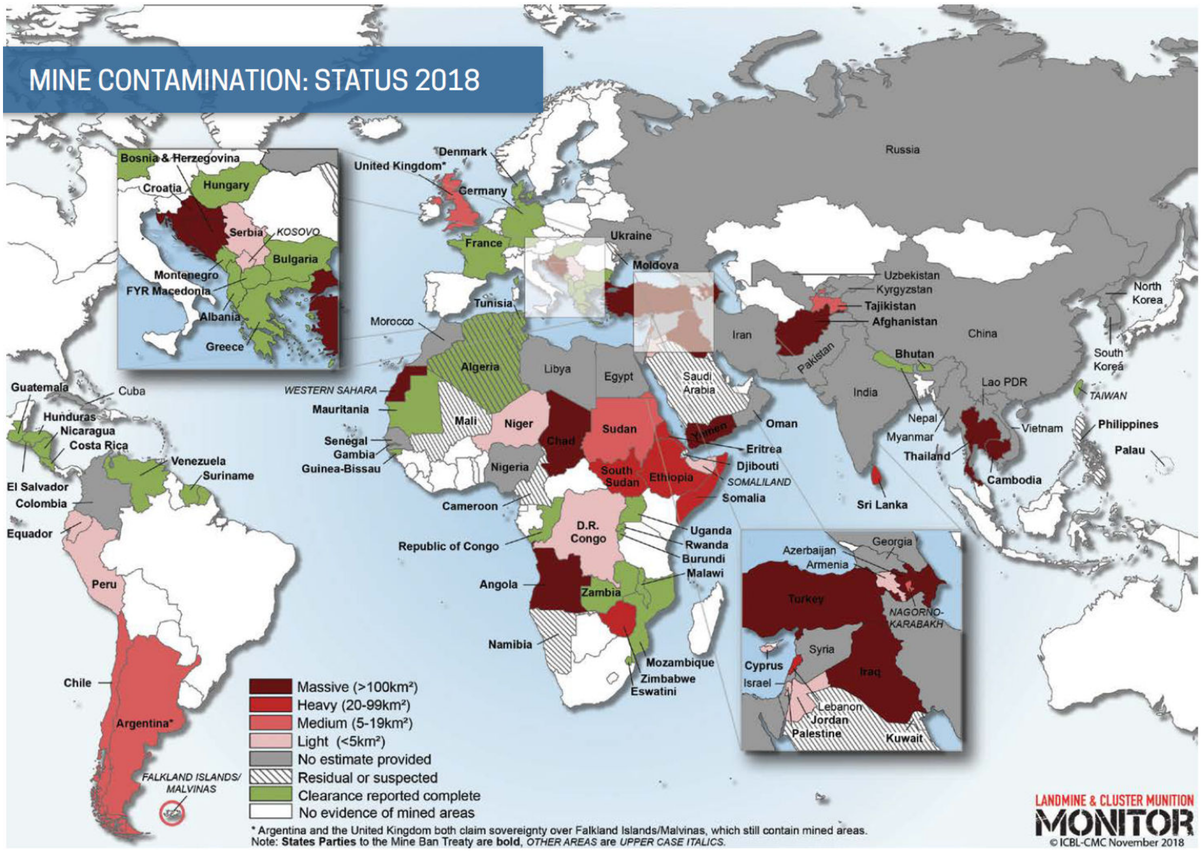


Figure 1.1 – Landmines, ERW and cluster submunition casualties in 2016. Reprinted [adapted] from 'Landmine Monitor 2017' by Monitoring and Research Committee, ICBL-CMC Governance Board, 2017, page 94.

a certain domain (for example: space, time, frequency, etc.) if it exists a basis (or frame) where this signal can be represented using only a very limited number of elements of the aforementioned basis. CS theory states that, if a signal can be sparsely represented, then it can be exactly recovered from a small set of linear, non adaptive measurements [16–21]. This implies, for example in radar, that we can reduce the sampling in time (i.e reducing Nyquist bandwith restrictions) or in space (i.e. less TX/RX elements) and still capture the essential information of the signal.

Even if we are not interested in reducing the sampling rate, one can find a basis where only a few coefficients are sufficient to represent the signal accurately (i.e. the signal is sparsely represented). This is a more general application of the same framework of techniques and it is called Sparse Representation (SR). The theoretical fundation of CS (and SR) was presented around 2006 by the brilliant contributions of Candes, Romberg, Tao and Donoho, however, the idea of recovering a signal from a smaller subset of measurements was already a challenge long before; the reader may find interesting to go through the pioneering works of Prony [22], Caratheodory [23], and, more recently: George et al. [24], Blu et al. [25] and Beurling [26].

Very briefly, SR and CS, deal with the solution of an under-determined system of equations where the number of measurements (samples) is generally smaller than the number of the constituting elements of a pre-selected basis. This basis is usually represented as a two-dimensional matrix which, depending on the different interpretations of the problems at hand, can be called *Sensing Matrix* or

Dictionary. The sensing matrix usually encompass a physical model and its columns (or *atoms*) can be seen as a collection of realizations of the signal of interest in a certain domain. CS applications that use this concept are aimed to optimize the selection of physical sensors to improve the estimation of the parameters of interest (for example the target position, speed, velocity, etc.) for a given scenario. The dictionary does not encompass any physical model, rather it contains a series of realizations of a given signal or its representation into another base. This is suitable for image processing and classification purposes where every realization (atom) could represent (for example) a particular patch of an image or a certain signal class associated to a particular target. For the purpose of this work we will refer to a dictionary instead of a sensing matrix.

The solution of the aforementioned system would lead to multiple results if sparsity in a certain domain is not assumed. Approaches which aim to solve this problem by limiting the number of non-zeros coefficients are called greedy algorithms; the most popular being Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), Block Orthogonal Matching Pursuit (BOMP), etc. [27, 28]. It has been demonstrated that approaches which solve convex optimization problems can also guarantee sparsity in the final solution [29], popular ones are Basis Pursuit and LASSO [30, 31].

When it is inefficient to pre-define the dictionary to contain arbitrary basis (e.g. Fourier, wavelets or empirically constructed ones), the usual resort is to *learn* the dictionary from previous measurements. Dictionary learning (DL) techniques aim to create adapted dictionaries which provide the sparsest reconstruction for given training-sets, i.e., a representation with a minimum number of constituting atoms [32]. Classical DL algorithms such as Method of Optimal Directions (MOD) [33] and K-SVD [34] operate in batches (Batch-DL) dealing with the entire training set in each iteration. Although extremely successful, these methods are computationally demanding and not scalable to high-dimensional training sets. An efficient alternative are Online Dictionary Learning algorithms (Online-DL thereafter) such as ODL [35], which converge fast, process small sets, and can infer the dictionary from large or time-varying training sets.

Radar applications can benefit from the use of CS and SR techniques [36]. The condition for using these techniques in radar is that the scene has to be sparse, meaning that there should be only a few dominant targets inside the observed domain. This condition can be easily fulfilled for detection and localization task where we have only a few point-like targets or applications where only a few extended objects/structures are present inside a low reflecting background. In these situations, one can use CS and reduce the number of measurements and still obtain a perfect reconstruction of the radar scene. Nowadays, only few real architectures integrate CS processing in the hardware [37], and only for experimental studies. Usually CS is applied in post processing and the full received signal is under-sampled before CS techniques are employed. A real CS architecture would, for example, directly generate less measurements at the transmitting stage. Among the most popular CS applications for radar there is pulse compression [38] where, respect to the classical matching filter approach, one can reduce the number of frequencies contained in the sensing waveform (i.e. a chirp) and still obtain an efficient estimation of the targets positions. As mentioned before, CS and SR can be used for radar target parameters estimation (such as range, angle and velocity), in this case the dictionary is replaced with the sensing matrix. The resolution of radar images can be increased over the Rayleigh limit (super-resolution) and the SNR may improve respect to classical beamforming (see figure 1.2), however, a tradeoff with computational burden and the increasing coherence of the sensing matrix must be made [36]. SR and CS are also used for direction of arrival estimation (DOA) of radar targets and their corresponding complex amplitudes. Normally we assume that range and

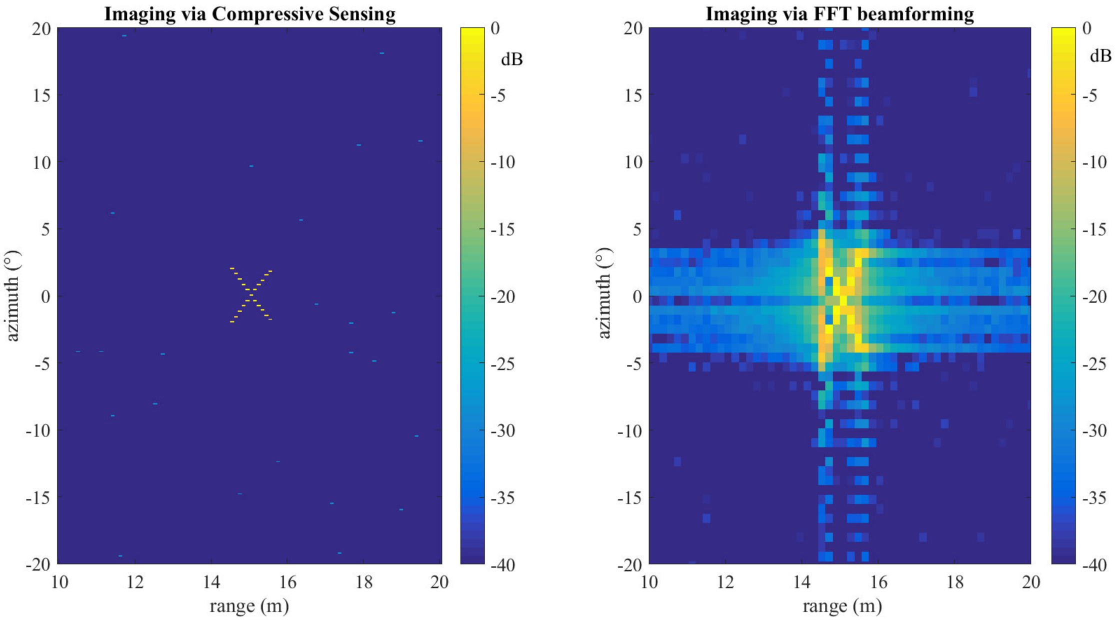


Figure 1.2 – Imaging of an X-shaped collection of point like targets using CS and FFT beamforming from a simulated Ka-Band MIMO radar. The SNR is 20dB and the CS image has been obtained using a factor sixteen undersampling of the original data. The resolution of the CS image is set to 0.1m in range and 0.1 degrees in azimuth, whereas the FFT beamforming has a resolution of 0.18m in range and 0.8 degrees in azimuth

Doppler estimation has been previously performed. The dictionary will contain all the possible steering vectors for a defined angular domain where the target may lie [39].

The application of interest for this work is sparse decomposition based classification (SRC). The idea of this approach is to use the vector of coefficients obtained from the application of SR as a collection of features for classification. For this purpose, one can use many signature signals which belong to different classes of targets (i.e. build a "dictionary of signatures") [40, 41] or one can use a collection of signals as the training set for a dictionary learning algorithm [42]. In both cases, the non-zero elements of the resulting coefficient vector represent discriminant features which can be associate to a certain signal class.

1.3 Major Contributions

The main scope of this work is to investigate the use of Online Dictionary Learning (Online-DL) techniques for SR based target classification for the case of abandoned anti-personnel landmines (APM), using experimental GPR data.

In particular we will highlight the benefits of using Online-DL techniques respect to Batch-DL approaches in terms of learning time and target identification accuracy. We will also introduce a novel Online-DL algorithm called DOMINODL which exploits the fact that a lot of the training data (for generating the dictionary) could be correlated and have similar sparse representation. The idea behind DOMINODL is to iteratively consider new and previous elements of the training set in small batches and dropping off the samples which become less relevant during the iterations. For assessing the performances and select the optimal parameters of the employed DL methods, we performed an evaluation which relies on statistical metrics such as Kolmogorov-Smirnoff test distance and Dvoretzky-Kiefer-Wolfowitz inequality. These metrics allow greater fine-tuning of parameters respect to the conventional bulk statistics such as root mean square error (RMSE). As an additional comparison with a state-of-the-art classification method, a Convolutional Neural Network architecture has also been evaluated. Finally, both CNN and DL-based classification outcomes are compared when the samples of the original range profiles are randomly reduced. Our experiments with real data from L-band GPR show that online-DL methods reduce learning time by 36-93% and increase mine detection by 4-28% over K-SVD. Our DOMINODL is the fastest and retains similar classification performance as the other two online DL approaches. Comparisons of the classification performance using CNN reveal that sparse decomposition based techniques with DL generally perform better than CNN alone when the input signals are randomly sub-sampled.

1.4 Thesis outline

In chapter 2 we will give a comprehensive overview on the GPR technology and GPR system architectures. We will then focus on GPR for de-mining applications and describe a work based on the parametric analysis of synthetic GPR signatures associated to landmine-like objects. A description of two classification algorithms used for GPR target recognition will also be provided in the appendices; those being Support Vector Machine (SVM, which will be an integral part of the proposed classification approach) and Convolutional Neural Networks (CNN) classifiers (which we will employ for comparison purposes).

Chapter 3 describes Sparse Representation (SR) and Dictionary Learning (DL) theoretical background and techniques. We will first introduce the the fundation of SR and describe the greedy and convex-optimization-based techniques that have been employed for this work. We will then introduce the Dictionary Learning theory, and provide an overview of state-of-the-art techniques for batch-DL and online-DL along with our novel DOMINODL approach.

In chapter 4 we present our DL-based classification approach in detail and assess its performance. We will provide accurate information on the employed experimental GPR dataset, describe an evaluation based on higher order statistical metrics (to select the optimal input parameters for DL) and finally analyze the classification results. Additional analysis will include the comparison of classification accuracy when using Convolutional Neural Networks and the influence of randomly reducing samples of the GPR signal in time.

Chapter 5 will then provide the conclusions and some insights for possible future works.

Throughout this thesis, we reserve boldface lowercase and uppercase letters for vectors and matrices, respectively. The ith element of vector $\mathbf{y}$ is $\mathbf{y}_i$ while the (i, j)th entry of the matrix $\mathbf{Y}$ is $\mathbf{Y}_{i,j}$. We denote the transpose by $(\cdot)^T$. We represent the set of real and complex numbers by $\mathbb{R}$ and $\mathbb{C}$, respectively. Other sets are represented by calligraphic letters. The notation $\|\cdot\|_p$ stands for the p-norm of its argument and $\|\cdot\|_F$ is the Frobenius norm. A subscript in the parenthesis such as $(\cdot)_{(t)}$ is the value of the argument in the tth iteration. The convolution product is denoted by $*$. The function $\operatorname{diag}(\cdot)$ outputs a diagonal matrix with the input vector along its main diagonal. We use $\Pr\{\cdot\}$ to denote probability, $\mathrm{E}\{\cdot\}$ is the statistical expectation, and $|\cdot|$ denotes the absolute value. The functions $\max(\cdot)$ and $\sup(\cdot)$ output the maximum and supremum value of their arguments, respectively.

Chapter 2

Ground Penetrating Radar Technology and Target Classification

In this chapter we will give an insight on the GPR technology, data acquisition and target classification strategies. In section 2.1 we provide some geophysical background to understand the complex scattering behavior of electromagnetic waves in the soil, focusing on the phenomenon that arises when probing the underground for shallow targets recognition.In section 2.2 we will explain the GPR basic principles and the different architectures of GPR systems, both in time and frequency domain. The visualization is a crucial step for interpreting the acquired data; thus we will explain the methodology to generate the so-called A-scans, B-scans and C-scans.

In section 2.3, the most relevant for our purposes, we will give an overview about GPR target classification. In subsection 2.3.2 we will describe a previous work about a parametric analysis of synthetic GPR signatures from landmine-like targets. This preliminary study was meant to understand how the target, soil and antenna characteristics will affect the GPR signatures of buried landmine-like targets, and therefore its correct classification.

In the appendices, we will describe two state-of-the-art classification methods which have also been used for GPR target classification: Neural networks (NN) and Support Vector Machines (SVM).

2.1 Geophysical Background

A GPR probes the underground by transmitting electromagnetic (EM) waves into the subsurface. In order to better understand this technology and its challenges, we will give a brief background on EM waves propagation and how these are affected by the electrical properties of the media [2, 4, 43, 44]. The reader is encouraged to follow the citations thorough this section to improve his understanding in the topic. Provided some boundary conditions are defined and assuming the general case of anysotropic (dispersive) media, the description of electric and magnetic fields vectors for a given point in space ($\mathbf{r}$ in $[m]$) and for a certain time instant (t in $[sec]$) is described by the *Maxwell equations* and the related *consitutive relations* in frequency domain:

$$\nabla \times \mathbf{E}(\mathbf{r},\omega) = -i\omega\vec{B}(\mathbf{r},\omega) \tag{2.1}$$

$$\nabla \times \mathbf{H}(\mathbf{r},\omega) = (\sigma(\mathbf{r},\omega) + i\omega\epsilon_0\epsilon_r(\mathbf{r},\omega))\,\mathbf{E}(\mathbf{r},\omega) \tag{2.2}$$

$$\nabla \cdot (\epsilon_0\epsilon_r(\mathbf{r},\omega)\mathbf{E}(\mathbf{r},\omega)) = 0 \tag{2.3}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r},\omega) = 0 \tag{2.4}$$

$$\mathbf{D}(\mathbf{r},\omega) = \epsilon_0\epsilon_r(\mathbf{r},\omega)\mathbf{E}(\mathbf{r},\omega) \tag{2.5}$$

$$\mathbf{B}(\mathbf{r},\omega) = \mu_0\mu_r(\mathbf{r},\omega)\mathbf{H}(\mathbf{r},\omega) \tag{2.6}$$

$$\mathbf{J}(\mathbf{r},\omega) = \sigma(\mathbf{r},\omega)\mathbf{E}(\mathbf{r},\omega) \tag{2.7}$$

where the vector **E** in $[V/m]$ is the electric field intensity, **D** in $[C/m^2]$ is the electric flux density, **B** in $[T]$ is the magnetic flux density, **J** in $[A/m^2]$ is the electric current density, ρ_c in $[C/m^3]$ is the electric charge density, ϵ_0 and μ_0 (in $[Farad/m]$ and $[Henry/m]$) are the electric and magnetic field constants, ϵ_r and μ_r (still in $[Farad/m]$ and $[Henry/m]$) are the relative dielectric permittivity and the relative magnetic permeability respectively, σ_r in $[S/m]$ is the electrical conductivity and the parameter ω in $[rad/s]$ is the angular frequency. The constitutive equations (2.5,2.6 and 2.7) do not generally hold for time-dependent fields (unless the medium of propagation is non-dispersive, such as vacuum), therefore, we choose to describe both them and the Maxwell equations (2.1,2.2,2.3 2.4) in frequency domain.

We assume the media to be homogeneous, this means that ϵ_r, μ_r and σ are indipendent of the position, though they generally remain complex and frequency dependent. However, for a typical GPR scenario with operating frequency from 10 MHz to a few GHz, the magnetic permeability can be neglected ($\mu_r = 1$), the imaginary part of the electric conductivity can be ignored and its real part can be assumed as frequency indipendent and equal to the DC conductivity [45].

The real part of ϵ_r (with $\epsilon_r = \epsilon_r' + i\epsilon_r''$) represent the electric permittivity of the soil and the imaginary part considers the losses for the conductivity and the frequency. For most GPR applications, the imaginary part of ϵ_r can be ignored and the real part is what affects the attenuation and the phase constant of the transmitted EM wave, resulting in a phase velocity (v_{p_h} in $[m/s]$) which is inversely dependent of the square root of the real part of the dielectric permittivity [4]:

$$v_{p_h} \approx \frac{c_0}{\sqrt{\epsilon_{\mathrm{r}}'}} \tag{2.8}$$

In typical GPR applications we are not dealing with an homogeneous media, rather we are in a situation where an EM wave reaches interfaces between different dielectric mediums. According to classical optical geometry, as indicated by Fresnel equations [46], depending on the dielectric properties of

MATERIAL	ϵ_r	σ (mS/m)	v (m/ns)	α (dB/m)
Air	1	0	0.3	0
Distilled Water	80	0.01	0.033	2e-3
Fresh Water	80	0.5	0.033	0.1
Sea Water	80	3e3	0.01	103
Dry Sand	3-5	0.01	0.15	0.01
Saturated Sand	20-30	0.1-1	0.06	0.03-0.3
Limestone	4-8	0.5-2	0.12	0.4-1
Shales	5-15	1-100	0.09	1-100
Silts	5-30	1-100	0.07	1-100
Clays	5-40	2-1000	0.06	1-300
Granite	4-6	0.01-1	0.13	0.01-1
Dry Salt	5.6	0.01-1	0.13	0.01-1
Ice	3.4	0.01	0.16	0.01

Table 2.1 – Dielectric constant (ϵ), conductivity (σ), velocity (v), attenuation (α) of different materials [44].

the materials, part of the incident energy will be reflected while the remaining will be transmitted to the lower medium. However, when we are facing a GPR scenario with shallow objects or interfaces in the close proximity of the illuminating source, the aforementioned equations are not sufficient to describe the EM waves behavior at the interface. Moreover, regarding the scattering of buried objects, optical geometry rules are valid only when the dimensions of the objects is much bigger than the wavelenght of the transmitting waves. When this situation is not fulfilled, one has to refer to the Mie or Rayleigh scattering methods (depending if the particles of the medium are respectively equal or much smaller respect to the wavelenght) to describe the EM waves propagation [47, 48].
The electrical properties of the soil are also affecting the GPR signal propagation and they depend on different factors: the volumetric water content, the frequencies of interest, the texture of the soil particles, the bulk density and the temperature [49]. Table 2.1 indicates the real part of the dielectric permittivity (the relative permittivity ϵ_r), the conductivity σ, the wave propagation speed v and the attenuation (α) for different materials that can be encoutered during a GPR survey.

Among all the components which contributes to the electrical permittivity of the soil, the water content is the one which affects the most. Many works proposed empirical and semi-empirical relationships between the moisture content and the permittivity [50–53]. Finally, another important physical phenomenons which affect EM propagation trough soils are the spatial variability and the roughness of the ground/air interface. The spatial variability can be expressed by the correlation lenght of the soil, i.e. the measure of the range over which fluctuations in one region of space are correlated with those in another region. The roughness of the Air-Ground interface influences the magnitude of the backscatter energy and depends on the surface characteristics and the wavelenght of the signal.

2.2 GPR principles

GPR operational principles are similar to the ones for conventional radar systems with the difference that now the waves are travelling into the subsurface. GPR systems can be employed for frequencies above 1MHZ, where the EM behavior is not inductive and can be described by EM waves (as indicated in the previous section). The penetration of the transmitted waves depends on the operating frequency, in general the lower the frequency the more the penetration in the subsurface. In other words, the backscattering from anomalies/targets which are much smaller with respect to the transmitted wavelenght is negligible whereas it becomes important for wavelenghts in the order of their size. However other factors such as the moisture content of the subsurface or, more generally, the dielectric properties of the soil can hamper the penetration depth. The attenuation coefficient (α) is defined as:

$$\alpha = \sqrt{\frac{\mu}{\epsilon}}\ \frac{\sigma}{2} \approx Z_0\ \frac{\sigma}{2\sqrt{\epsilon_r}} \tag{2.9}$$

Where ϵ is the permittivity of the soil, μ is the magnetic permeability and σ is the electric conductivity, as for Z_0 in $[\Omega]$ we refer to the EM impedance defined as:

$$Z = \sqrt{\frac{\mu}{\epsilon}} \approx \frac{Z_0}{\sqrt{\epsilon_r}} \tag{2.10}$$

with

$$Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}} \tag{2.11}$$

being the EM impedance and $\epsilon_r = \frac{\epsilon}{\epsilon_0}$ being the relative permittivity when neglecting $\mu_r = \frac{\mu}{\mu_0}$. Figure 2.1 shows the penetration depth of a GPR in function of the frequency, for various types of materials. The aforementioned figure indicates that the penetration depth changes also in function of the dynamic range; an upper and lover operation curve is shown for each material, forming a boundary between 150 and 100 dB of dynamic range. The dynamic range of a GPR system indicates the difference (in dB) between strongest and weaker signal which can be recorded; these limits should bracket the ground reflection and the noise level of the system [5]. The dynamic range defines the capability to detect weak signals in the presence of strong superimposed signals (i.e. the ground reflection) and it is directly correlated to the penetration depth capability [54]. For many applications such as landmine detection, there is a need to work with higher frequencies (i.e. from 1GHz) to obtain the necessary resolution for discriminating very small targets; in these situation the penetration depth decreases dramatically [4]. GPR system usually operate between 30MHz to 3GHz.

The speed of the EM wave into the soil is also affected by the dielectric properties, in general the more the dielectric permittivity gets higher, the more the velocity decrease. We define the phase velocity of the EM wave into the subsurface (in m/s) as:

$$v_{ph} = \frac{1}{\sqrt{\epsilon\mu}} \approx \frac{c_0}{\sqrt{\epsilon_r}} \tag{2.12}$$

where,

$$c_0 = \frac{1}{\sqrt{\epsilon_0\mu_0}} \tag{2.13}$$

being the phase velocity in free space with $c_0 = 3x10^8$ m/s.

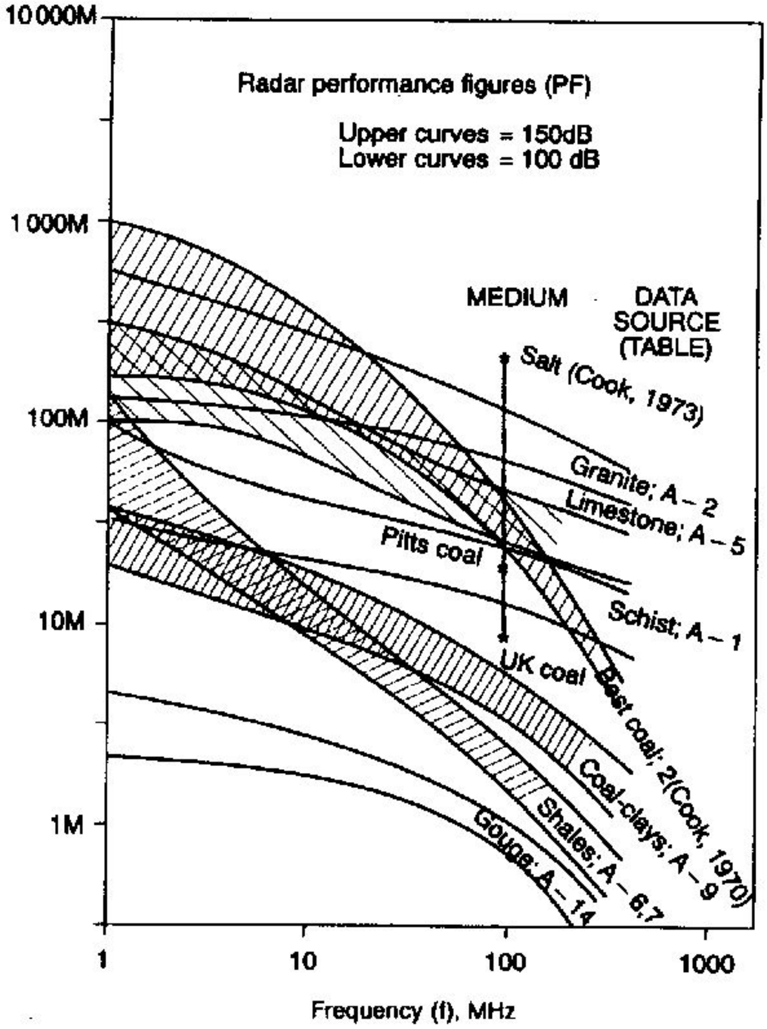


Figure 2.1 – Penetration depth vs. frequency of operation for different soil types. Reprinted [adapted] from 'Surface-penetrating Radar' by D.J. Daniels, 2004, page 19.

The radar reflections happen when there is a change in the EM properties of the soil (in particular the dielectric permittivity). The reflected signals will contain the contribution of a specific target at a certain depth, plus all the unwanted effects from antenna coupling, system ringing and interface/soil contributions.

According to the operation mode, GPR systems can be classified in time-domain or frequency-domain GPR. Systems that transmit a short time pulse and receive the backscattered reflection via a sampling-based receiver operate in time domain while systems that transmit a series of individual frequencies and receive the signal via a frequency conversion receiver operate in frequency domain. We are going to describe only the architecture of time-domain pulsed GPR systems like the one which has been used for this work (see chapter 4).

Pulsed GPR systems transmit a series of short pulses (with duration between 200ps and 50ns) with a repetition interval which goes from hundreds of microsecond to one millisecond. UWB GPR (like the one used in the experiments described in section V) send pulses with central frequency (f_c) between some MHz up to 1 GHz and a bandwidth which is equal to f_c. These kind of GPR systems are used for shallow target detection (such as landmines). The resolution of a GPR in depth (Δ_z in $[m]$, when the depth axis is z according to cartesian coordinate) is calculated like this:

$$\Delta_z = \frac{v_{ph}}{B} \tag{2.14}$$

As we can see, the resolution depends on the phase velocity, thus, it also depends on the dielectric permittivity of the material where the target lies. As an example, a radar with receiver bandwidth of 2GHz can resolve targets which are spaced from 10cm to 1cm apart from each other, depending on the value of ϵ_r. The main architecture of an UWB pulsed radar is shown in figure 2.2. The pulse generator uses a technique of rapid discharge of the stored energy (from the high voltage supply) into a short transmission line for generating the pulse, this approach gets more complicated when increasing the pulse repetition time. The receiver quantize the signal using a sample/hold technique then uses an flash ADC for digital conversion. Dipoles and Bow-Tie antennas are widely used in pulsed GPR since they are wideband, easy to design, non-dispersive and linearly polarized [55].

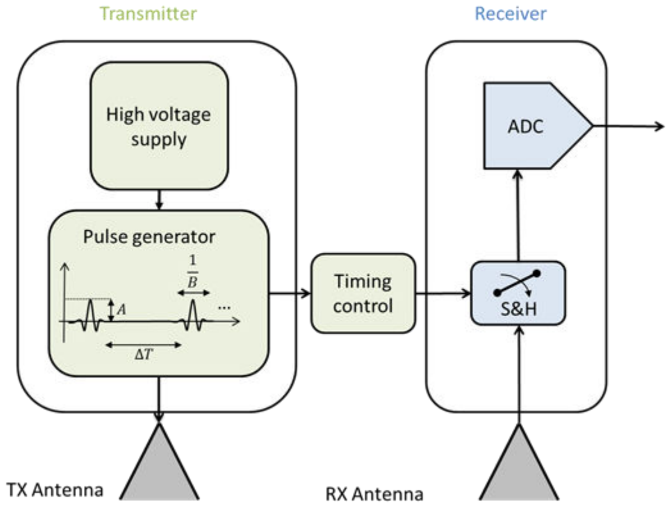


Figure 2.2 – Pulsed UWB GPR system architecture

Usually they are sealed in a shielding box filled with absorbing material to prevent coupling issues. A descriptive representation of how an UWB pulsed GPR works is shown in figure 2.3.

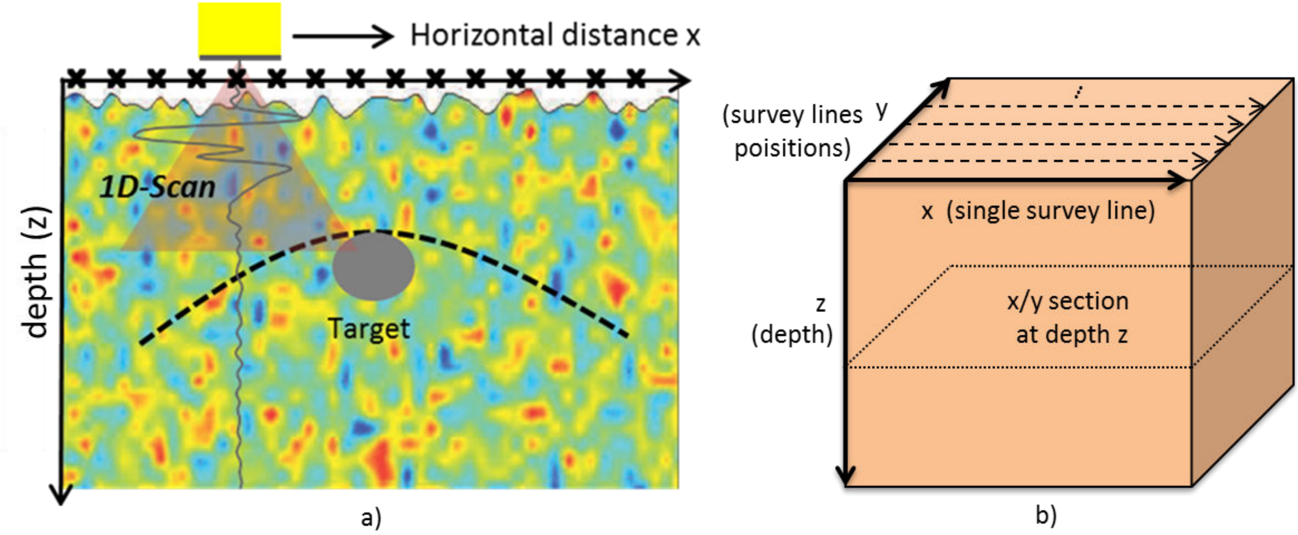


Figure 2.3 – a) Down looking GPR system operation indicating a single range profile b) Direction of a single survey line (x), survey lines positions (y) and depth (z) in 3D, indicating a 2D section at depth z (C-Scan)

The GPR moves along a survey line and, for each horizontal position (indicated with crosses), sends a pulse (or a series of pulses) and record the backscattered reflection (or integrate a series of received ones). The GPR is usually moved by a single operator or pulled by a veichle. In our experiments, the GPR was moved along pre-defined tracks by the means of a precise mechanical system. The acquisition is usually triggered in function of the traveled distance (for example by an odometer connected to a wheel). For each horizontal position, we obtain a single range profile (also called 1D-scan or A-scan as shown in figure 2.3) usually generated from the integration of many received pulses (to increase SNR). If we stack many range profiles together along the survey line, we obtain a 2D visualization (also called B-Scan, see figure 2.4a). Due to the beamwidth of the antenna, the GPR will start receiving reflections from a target before he is right above it, this result in the well-known hyperbola-like response in the B-scan where the edge of the hyperbola is associated to the real depth of the target. If the acquisition has been repeated for many survey lines, we could stack them togheter to form a *datacube*. In figure 2.4b we show a X/Y section of the aforementioned datacube for a certain depth, this visualization is also called C-scan.

2.3 GPR target classification

GPR classification approaches aim to discriminate target/anomalies from the ground clutter, possibly indicating their precise position in the analyzed surveys. Unlike GPR imaging approaches (such as backprojection, migration/focusing etc.) which aim at improving the visualization of the targets in the final GPR image, the output of classification methods can be seen as a map of declared classes along the survey area (see an example from experimental measurements in figure 2.5).

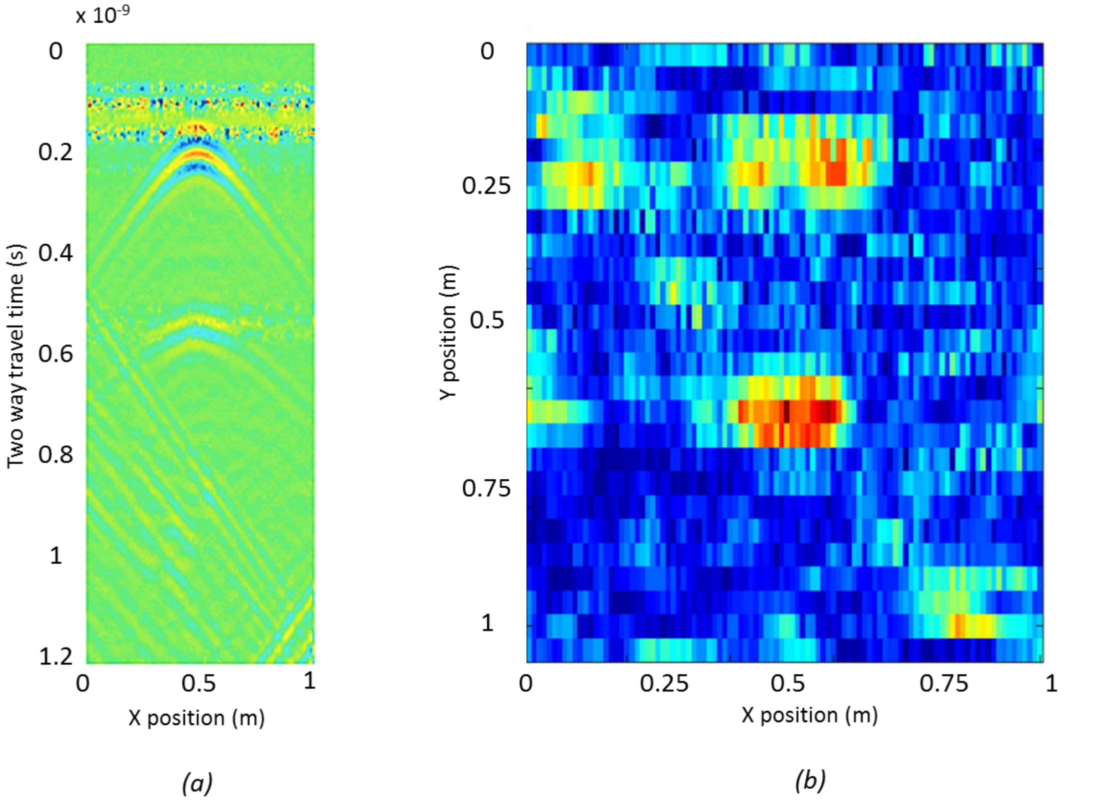


Figure 2.4 – a) B-scan of a simulated spherical target buried in sand material b) C-Scan from real data at 15cm depth of two landmine simulant buried in highly non-homogeneous soil

2.3.1 Overview and challenges

GPR classification approaches can work on raw or pre-processed data, in time or frequency domain, and can evaluate single A-scans or a collection of received GPR data (such as entire B or C-scans). Many classification approaches rely on a database of GPR signatures. This can be synthetically generated (for example, by a modeling software like GprMax [56] or Comsol) taking in account all the necessary parameters (soil, antenna, transmitted signals, etc) of the scenario under test. However, having a complete and general database of GPR signatures is usually very challenging; in many cases it is more convenient to extract salient features from a *representative* database of the target of interest and use them as an input for the chosen classifier. A representative database is smaller w.r.t. a general database since it contains only an accurate selection of the classes of interest for the desired application.Among some GPR signal features that demonstrated to be promising for buried targets/structures discrimination purposes we indicate Wavelets [41, 57, 58], Target resonances [8, 59], Top/bottom reflection ratios [8, 41], Edge histogram descriptors [60] and Sparse representation coefficients [40]. Figure 2.6 schematically summarizes the typical approach for GPR target classification.

The features that have been used in this work are Sparse Representation Coefficients; their efficient

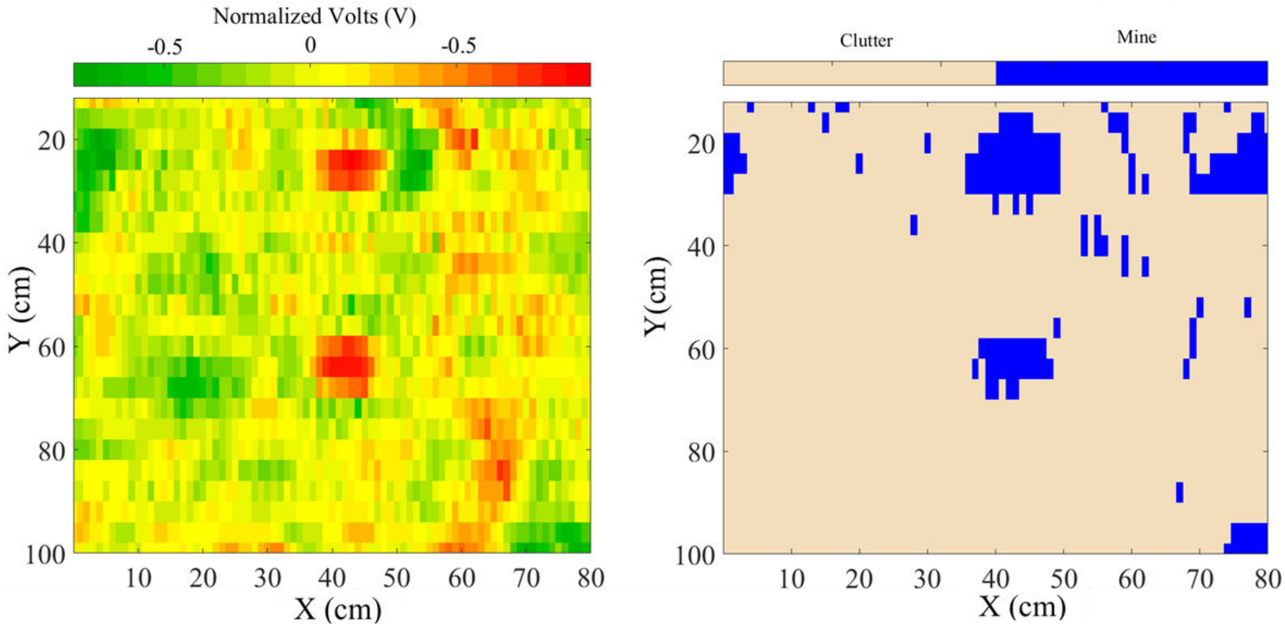


Figure 2.5 – Example of a classification map (right) of a GPR survey area which contains 2 landmine simulants. On the left side we show the related raw data (from experimental measurements) at 15cm depth (left), here a sparse representation based classification approach was used

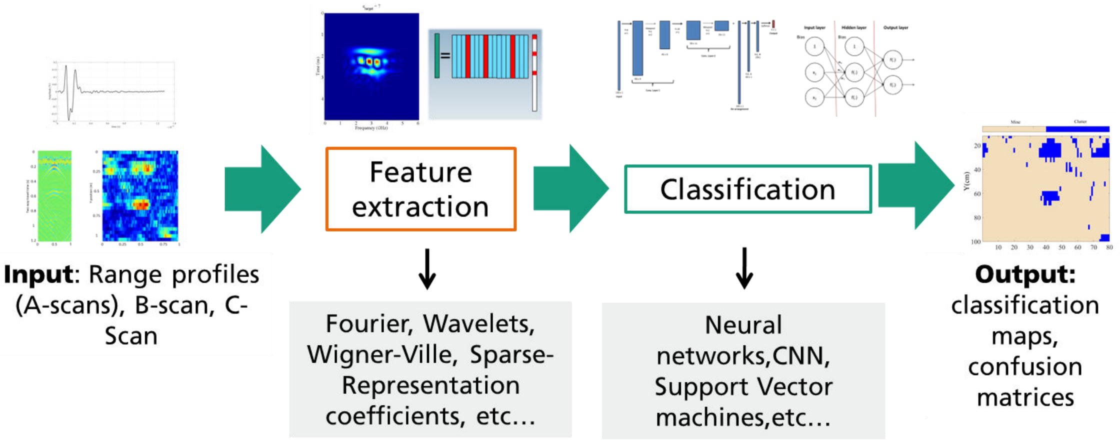

Input: Range profiles (A-scans), B-scan, C-Scan
Feature extraction
Fourier, Wavelets, Wigner-Ville, Sparse-Representation coefficients, etc...
Classification
Neural networks,CNN, Support Vector machines,etc...
Output: classification maps, confusion matrices

Figure 2.6 – GPR target classification, general approach

extraction is the key for the success of the proposed classification strategy (see chapter 4).

As mentioned in chapter 1, the data received from a GPR system is subject to many sources of noise and/or unwanted phenomenons which hinder the reflections of buried targets and the subsurface features of interest, thus making classification tasks very difficult. Below, we indicate some of the effects which could hamper the correct classification of GPR signals:

Ground reflection Being the antennas of GPR systems normally standing at some cm above the surface, one has to take in account a considerable reflection from the air/ground interface which in most cases is stronger than any other reflection.

Antenna cross-coupling The cross coupling between the receiver and the transmitter antenna cause the transmitter signal to be harmfully interfering with the received one even though absorber materials are commonly used in between of TX and RX modules.

Antenna ringing Antenna ringing effect cause the transmitting antenna to continue radiation after the exciting source has expired. In a Bow-tie antenna (one of the most popular design for GPR applications) this is due to the internal reflections of the charges that occur at the ends of the bow-tie wings and at the excitation points, this corrupts the transmitted pulse waveform and reduces range resolution.

Surface roughness Depending on the frequency of exploration, in accordance with the Rayleigh criterion, the roughness of the surface can cause undesired scattering patterns which are not associated with subsurface features [61]. This is especially true for shallow surface exploration (i.e. for landmine detection) where the frequencies of operations are around 1-2 GHz.

Moisture content The higher the moisture content in the subsurface the more the signal is attenuated during its propagation in the subsurface [4]. The attenuation due to water content depends on the frequency of operation and its strictly related to the dielectric permittivity of the soil.

Low permittivity contrast The intensity of reflections received from the GPR depends on the permittivity contrast between the soil where the EM wave is propagating and the targets/features of interest [2]. Small, Non-metallic targets (such as landmines) usually have very low permittivity contrast respect to the soil, making its discrimination extremely challenging.

To deal with these issues, pre-processing on the raw GPR data can be necessary. Here we present a short list of just some of the popular pre-processing techniques for time-domain pulsed GPR.

Time gating Time gating consists in cutting the GPR data in the range/time dimension, thus limiting the prospection depth between a specified interval. In some situations, this practice is useful to remove the effects from the ground reflection and/or limit the data only to depths where targets returns are expected. Unfortunately, cutting the ground reflection is not always beneficial since some shallow targets (i.e. landmines) responses mix up with the reflection from the air/ground interface. Time gating can be also used in post-processing.

Dewow Dewow is a running average filter to remove the initial DC and low-frequency component. A common way to perform dewow is is to calculate the mean among some GPR profiles (A-Scans) taken along a section then subtract this mean value from every single profile in the section. The length of the section depends on the wavelength/frequency of the signals.

Time gain Some GPR architectures can incorporate a time-varying gain to compensate from the attenuation of the signal during its propagation trough the ground. The gain curve can be customized in order to account to a more smooth or edge transition along the depth. Time gain can also be applied in post processing to improve data visualization.

Background subtraction Background subtraction is commonly implemented by subtracting the mean profile (over an entire survey line) from each profile in the line. It is different from Dewow since it takes an entire survey line instead of a "moving window". A popular way for performing background subtraction is via Pricipal Component Analysis (PCA) [62, 63] . In the latter, singular value decomposition is perfomed to extract the first principal values of the data, which are associated to the background, and subtract their contribution from the considered profiles.

Thanks to numerical modeling it is possible to simulate complex physical phenomena via dedicated tools. For the application of landmine recognition with GPR, an accurate simulation of a GPR scenario (including the antenna, the soil and the target) becomes important to test feature-extraction and target-classification algorithms. Moreover, the obtained results can be employed to build a database of synthetic outputs, which may be used for example as a training set for automatic classification algorithms.

2.3.2 Analysis of GPR scattering signatures from landmine-like targets

In order to better understand the scattering mechanisms of buried APMs in response to GPR signals and to identify discriminant characteristics for classification, a comprehensive parametric study is useful and made possible by synthetic modelling of a typical GPR scenario for landmine clearance.
In this subsection we present an extensive study of Ultra Wide Band Ground Penetrating Radar returns by small buried targets using electromagnetic modeling software [8]. There are several successful GPR modeling attempts reported in specialized literature and many of them are based on the Finite Difference Time Domain (FDTD) method. This numerical technique works by assigning appropriate constitutive parameters to a collection of defined discrete locations (cells) that made up a given computational domain where the electromagnetic field components are operating. The modeling software used in the presented work is GprMax V2.0, which is freely available on the web [56]. Our model for the parametric analysis consists of a parallel pair of bow-tie antennas (one transmitter and one receiver) enclosed in a shielding case (with or without filling absorber material) and placed 5cm above a typical sandy soil ($\epsilon_{soil} = 5$). The transmitted pulse is a Monocycle (or Ricker) type with a central frequency of 2GHz (see fig. 2.7). Given the Gaussian waveform

$$s_G(t) = Ae^{-2\pi^2 f_c^2 (t-\tau)^2},\ t \in [0, +\infty], \tag{2.15}$$

where f_c is the central frequency, A is the peak amplitude and $\tau = 1/f_c$, the monocycle waveform is its first derivative [64]

$$s_T(t) = -4\pi^2 f_c^2 A(t-\tau)e^{-2\pi^2 f_c^2 (t-\tau)^2},\ t \in [0, +\infty]. \tag{2.16}$$

In these UWB systems both the central frequency and the bandwidth are approximately the reciprocal of the pulse length. The reference landmine-like target is a plastic cylinder ($\epsilon_{target} = 3$) of 10cm diameter and 5cm height. The whole GPR model is shown in fig. 2.8.

In our simulations we analyze single received 1D traces (A-scan), computed by placing the GPR antennas a few centimeters above the ground and the targets at a specific depth in soil and directly below the antennas. The aforementioned responses can be studied in either time-domain (TD), frequency domain (FD) or considering functions of both variables, time and frequency, i.e., time-frequency domain analysis (TFD).

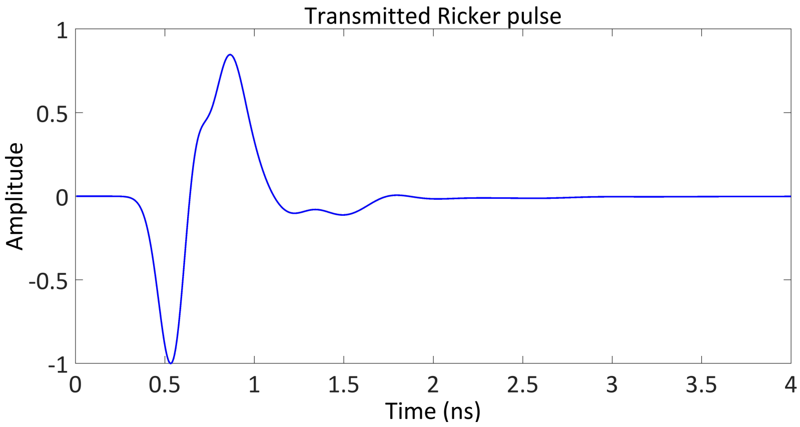


Figure 2.7 – Simulated transmitted Ricker pulse with Bow Tie antennas enclosed in a shielding metal box filled with absorber material

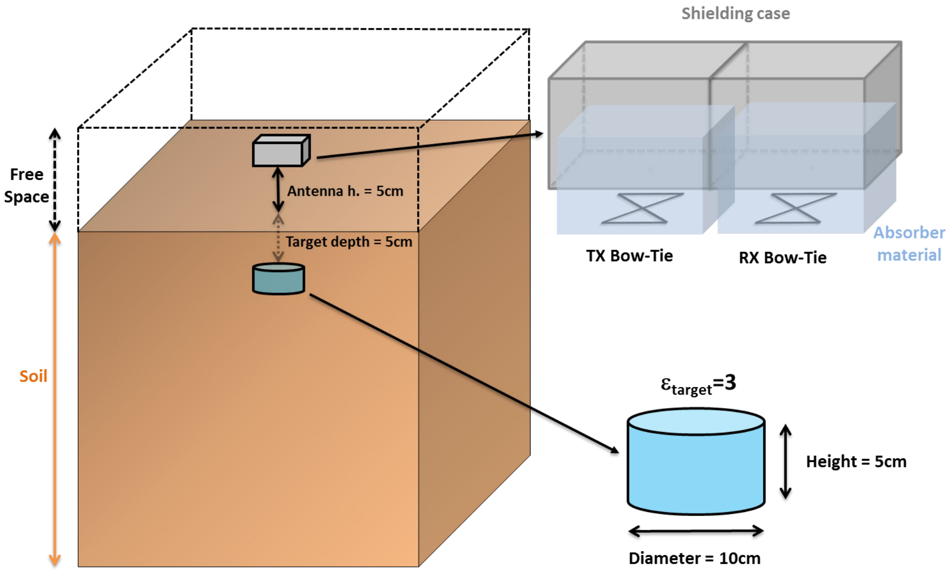


Figure 2.8 – The modeled GPR scenario.

2.3.2.1 Time domain analysis

TD analysis is probably the most natural descriptor of a given signal. In our application, the recorded signal in time is the sum of the contributions coming from the buried object, the surface and the soil plus the direct coupling between the transmitter and receiver antennas or crosstalk (that coincides with the first arrival). Hence, to isolate the response of the target, we subtract the received signal in the simulation without target (background) from the total received signal in the simulation with the buried target. In the following analysis we mostly refer to the Hilbert signatures which correspond to the analytical signal obtained via Hilbert transform of the real-valued temporal signal s(t). The shape (or certain features extracted from it) of these signatures may be successfully employed as target classifier. We will also analyze here the behavior of the first and second main peaks present in the signature.

2.3.2.2 Frequency domain analysis

FD analysis essentially describes which frequencies are contained in the waveform, as well as their corresponding amplitudes and phases. Considering a signal as a group of contributions recorded with time, its spectral response (using the Fourier Transform) shows how much of $s(t)$ lies within each given frequency over a range of frequencies. In order to investigate the energy concentration in the spectral response as a potential discriminant, we determine the number and position of resonances present in the FD signature for several scenarios.

2.3.2.3 Time-Frequency domain analysis

Time-frequency distributions, which are 2D functions, can reveal the time-varying frequency content of 1D signals. A simple way to characterize a signal simultaneously in both domains consists in distributing the energy of the signal along a time-frequency space. To apply such approach, we can define the short-time Fourier transform (STFT) and define the so called spectrogram [65]. However, the resolution of the distribution is limited by the STFT. A possible approach to enhance the resolution is the wavelet transform (WT) [57], which is characterized by the ability to segment the time-frequency plane into localized regions of interest. Nevertheless, a non-trivial point in the WT is the appropriate selection of the basic wavelet. Another 2D distribution which shows a better time-frequency resolution than the STFT and the continuous wavelet transform can be obtained via the Wigner-Ville transform or Wigner distribution function (WDF) [65]:

$$WD(t,\omega) = \int_{-\infty}^{+\infty} x(t+\frac{\tau}{2})x^*(t-\frac{\tau}{2})e^{-i\omega\tau}d\tau \tag{2.17}$$

where $x(t)$ is the analytic associate of the real-valued time signature, τ is the time shift and ω is the angular frequency. The WDF does not require the selection of additional time–frequency functions or wavelets. We have selected this sort of time-frequency transform for the present study and we will show some examples of the resulting time-frequency distributions assuming different target scenarios. The obtained real matrices can provide some discriminant features between targets.

2.3.2.4 Study of the scattering responses

We compared different signatures in time and frequency domain when changing different parameters in the presented GPR scenario, the results are shown in figures 2.9-2.11 and commented below.
Antenna Height (Fig. 2.9a): Modifying the antenna height above the ground changes the shape and amplitude of both time and frequency signatures. Basically, when we elevate the antenna the overall amplitude of the target echo decreases due to a worse energy coupling into the soil. Nevertheless, for landmine applications, it is necessary to keep the antenna at a certain distance from the ground for safety reasons and to avoid bounces when moving the system in a wrinkled terrain. A good tradeoff is achieved by placing the antenna at 5cm height. The duration of the time domain signature is dependent on the antenna height and it seems larger when we have a good energy coupling into

the soil (i.e. h=1cm) but it does not change very much between the 3 and 5cm cases. Accordingly, the resonances and the overall shape of the frequency signature changes mostly for 1cm height.

Target Shape (Fig. 2.9b): The target shape affects importantly to the profile of the time and frequency signatures. More specifically, there is a drastic variation on the number and position of resonances in frequency domain. Regarding the magnitude, the echo coming from the sphere has a smaller amplitude than the ones produced by the cube and the cylinder; this is due to its smaller RCS (Radar Cross Section).

Target horizontal dimension (Fig. 2.9c): Keeping the cylinder height fixed at 5cm (vertical dimension) and increasing the diameter of the cylinder (horizontal dimension) results in an increasing magnitude of both the time and frequency domain signatures. When the horizontal dimension is significantly bigger than the vertical dimension (diameter range: 10-15cm) the signature shape does not change very much. On the other hand, for smaller or similar horizontal dimensions, there is a significant variation. The frequency signature is affected as well, in particular the resonances get more separated from each other when increasing the diameter.

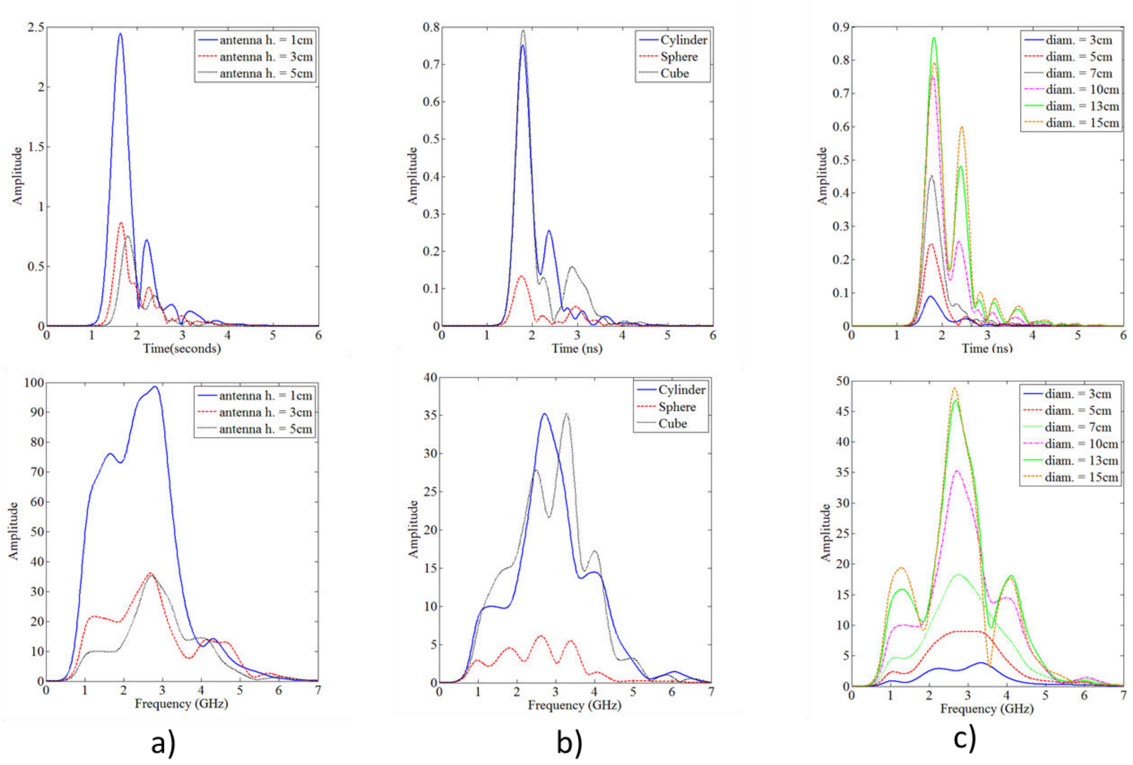


Figure 2.9 – Signatures in time (upper plots) and frequency (lower plots) domain of a plastic cylinder buried in a sandy soil (5cm depth) with a) different antenna heights, b) different target shapes and c) different cylinder diameters.

Target vertical dimension (Fig. 2.10a) : Increasing the height of the cylinder and keeping its radius fixed causes the second reflection of the Hilbert signature to shift and decrease in magnitude while the first reflection remains unchanged. This is because of the increasing travelling time and attenuation through the object when it gets thicker, i.e. the reflections from the bottom of the cylinder arrive later. Frequency signatures are changing a lot without a clear trend.

Target depth (Fig. 2.10b): As expected, the signatures in time domain shift and decrease in amplitude when increasing the depth of the cylinder. The shapes remain very similar (at least for negligible soil conductivity values). Accordingly, the amplitude of the frequency signature decreases in amplitude with the depth and its shape and the resonances are not very sensitive either apart from a slight shift for the 7cm and 10cm case.

Target inclination (Fig. 2.10c): The rotation of the cylinder modifies his apparent RCS to the incoming wave, and hence the backscattered signal. We observe that, both for time and frequency domain, there is almost no difference in rotating the cylinder clockwise or counterclockwise and along x or y axes; this is due to the symmetry of our model. Concerning the time domain, when we rotate the cylinder 45°, the signature is highly attenuated and its shape changes appearing some ringing effects; a rotation of 90° results in an attenuation of the amplitude as well but with less ringing. Accordingly, the frequency signature changes in shape and magnitude with the rotation angle, in particular the number of resonances increases significantly with a rotation of 45°.

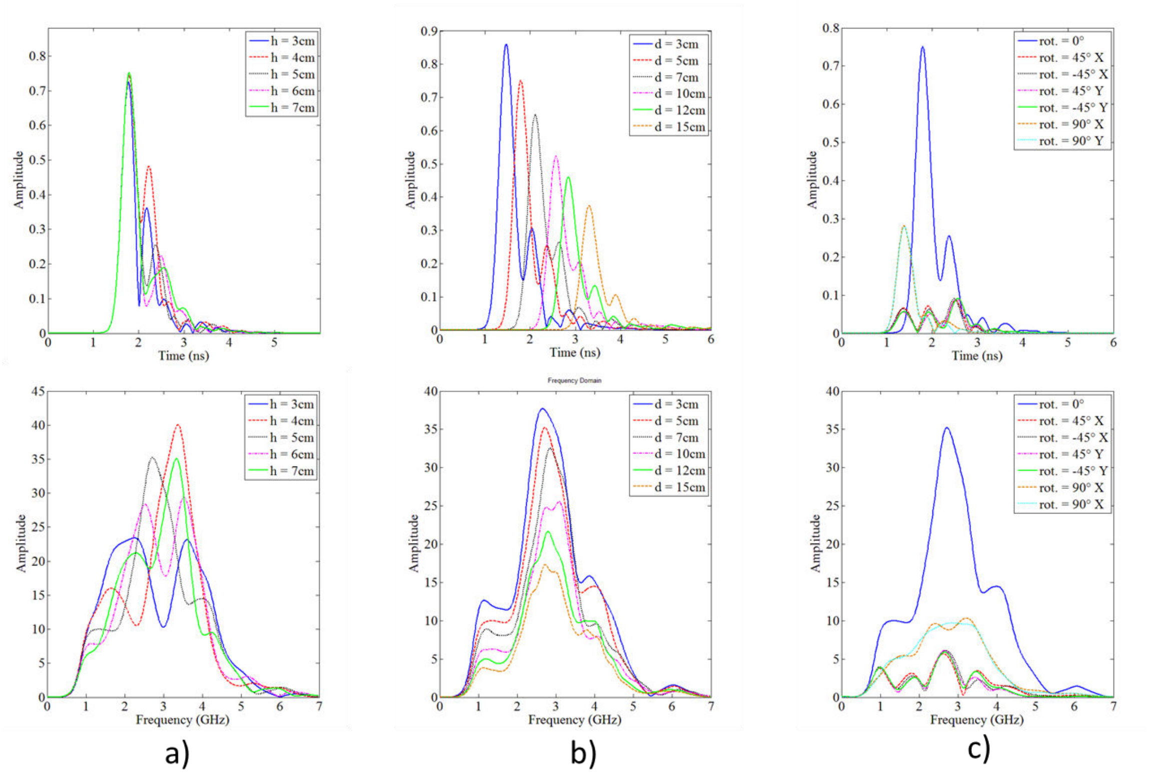


Figure 2.10 – Signatures in time and frequency domain of a plastic cylinder buried in soil with a) different cylider heights, b) different cylider depths and c) different cylinder inclination.

Target permittivity (Fig. 2.11a): The dielectric properties of the target affect the signatures remarkably. For this analysis we can consider the signature in terms of the permittivity contrast between the target and the soil.

$$Contrast = \frac{\epsilon_{target} - \epsilon_{soil}}{epsilon_{soil}} \tag{2.18}$$

where ϵ_{target} and ϵ_{soil} are respectively the relative dielectric permittivity of the target and the soil. In the obtained signatures it is usually possible to locate two main contributions that are associated to the wave reflected at the top interface of the target and the reflection coming from the bottom interface (both peaks are indicated with arrows in Fig. 2.11a). Note that the secondary contributions present at the backscattered responses by plastic targets (which are of similar dimension as the illuminating wavelength) are due to different multiple scattering mechanisms (in addition to the bottom reflection), resulting from the wave travelling inside and along the object [15].When the target permittivity value increases above the one of the soil, the bottom reflection shifts in time and its contribution becomes comparable to the one from the top reflection. From the figure we also see that the sign of the contrast is important (i.e. $\epsilon_{target} = 3$ vs. $\epsilon_{soil} = 5$ do not generate the same signature as $\epsilon_{target} = 7$ vs. $\epsilon_{soil} = 5$), i.e., there is not a "symmetric" scattering behavior. When the permittivity of the target grows, the frequency signature presents an increasing number of resonances.
Soil permittivity (Fig. 2.11b): Changing the permittivity of the soil and maintaining the target one constant results in a shift of the top reflection depending on the value of the soil permittivity (due to the different travelling velocity in different materials). Top/bottom amplitude ratios vary with the contrast but obviously the bottom reflection position does not shift since the target permittivity remains the same. In frequency domain we still can see that increasing the contrast always produce more resonances and as expected, raises the magnitude of the signature.
Soil conductivity (Fig. 2.11c): The electrical conductivity affects only the magnitude of the signature while the shape remains mostly unchanged; this is confirmed for both time and frequency domain. More precisely, when we increase the electrical conductivity, the attenuation of the travelling wave grows and consequently the amplitude of the signature decreases. The electrical conductivity of the soils depends mainly on their water content (along with salts and clay content).

Some Wigner-Ville distributions corresponding to the time signatures of the reference cylinder for varying depth, antenna height and target permittivity are displayed in Fig. 2.12. As we can see, the different depths mainly shift the position in time of the WVD without altering its overall shape and intensity, while the height and in particular the permittivity of the cylinder, affect significantly the whole appearance of the target image. We are not going to make a deeper analysis here, but it has been shown in literature that landmine discrimination based on WV time-frequency signatures may be successfully performed by the extraction of certain features via singular value decomposition and principal component analysis [63].

We present here some results associated to the variation of the analysis parameters in the model in order to have a better understanding of the impact of each investigated factor. We will also analyze two descriptors extracted from the data: the top/bottom amplitude ratio and the number and position of resonances, both in function of the permittivity contrast and the shape/dimension of the target. In table 2.2 we show the correlation coefficients between different GPR signatures for different values of model parameters. The correlation coefficient C is calculated as follows:

$$C = max(\overline{|r_{xy}(\tau)|}) \tag{2.19}$$

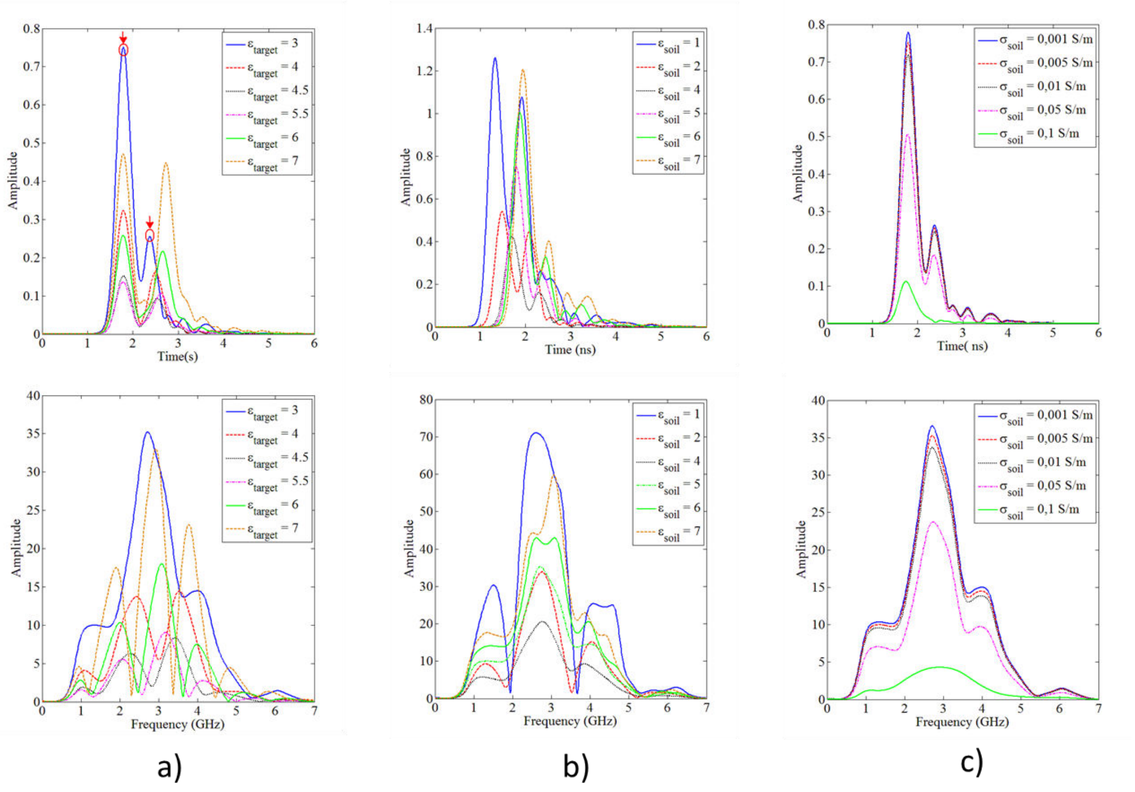


Figure 2.11 – Signatures in time and frequency domain of a plastic cylinder buried in soil with a) different target permittivities, b) different soil permittivities and c) different soil conductivities.

where:

$$\overline{r_{xy}(\tau)} = \frac{r_{xy}(\tau)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \tag{2.20}$$

is the normalized cross-correlation function with:

$$r_{xy}(\tau) = \int_{-\infty}^{+\infty} x(t)y(t+\tau)dt \tag{2.21}$$

being the cross-correlation function i.e. a measure of the similarity between two different signals ($x(t)$ and $y(t)$) as a function of the time shift τ between them. Each entry of the table 2.2 corresponds to a value of the correlation coefficient between signatures generated from target with different height, depth, diameter, permittivity, rotation and soil permittivity. The similarity results are consistent with the analysis made in the previous section. The depth of the cylinder does not produce a significant variation, while the dimensions (diameter and height) and the target permittivity are remarkably affecting the echoes. It is also interesting to remark that the values associated to the rotation of the target are in agreement with what we explained in section 3 except for the 90° rotation which appears similar to 0° rotation. Finally, the soil permittivity does not seem to have very much impact on the correlation coefficient. Figure 2.13 shows the variation of the top/bottom amplitude ratio in function of the permittivity contrast. The top/bottom amplitude ratio is just the amplitude of the bottom

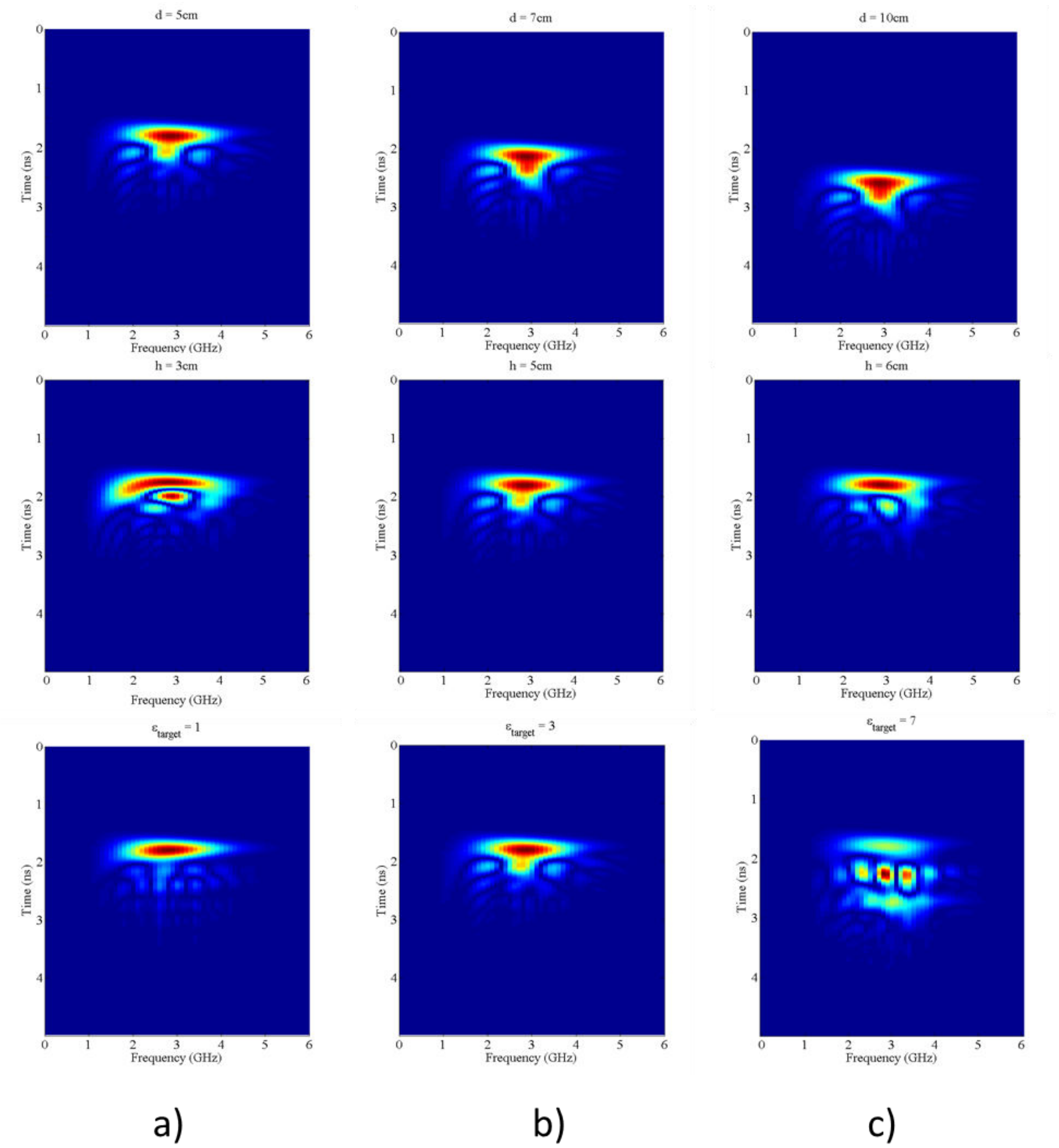
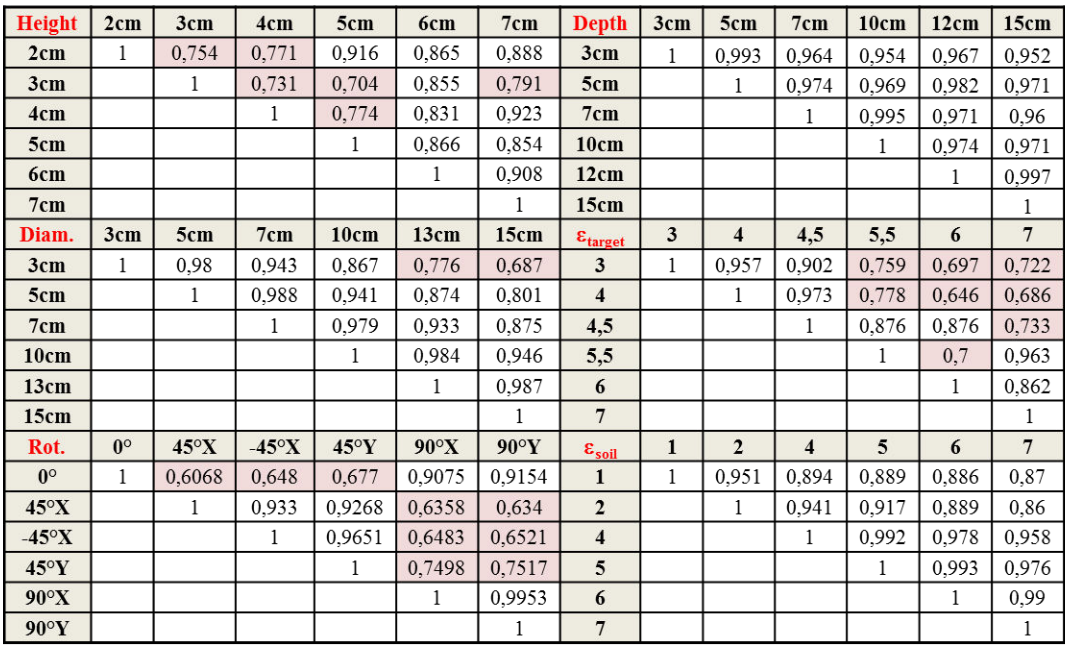


Figure 2.12 – Wigner Ville transforms of the signatures from a cylindrical target buried in sandy soil at different depth (a), different height (b) and different permittivity (c).

reflection divided by the amplitude of the top one. To locate these values, we automatically select the peaks that agree with the estimated arrival times of the top/bottom reflections from the Hilbert signature of a given target. We investigated this ratio for a fixed $\epsilon_{soil} = 5$ (sand) and different values of ϵ_{target}, assuming different shapes (sphere, cylinder) and dimensions of the targets. It is interesting to see that, for all the geometries, there is a general decreasing trend of the ratio when the contrast increases, even if this is clearer for values starting from -0.5 (which means $\epsilon_{target} = 3$). In fact, the cylinder with diameter equal to 10cm has a different behavior for values of the contrast below -0.35. Some values of contrast are not covered because we couldn't distinguish any bottom reflection. This happens for example for the case of the cylinder with diameter 5cm and $\epsilon_{target} \approx 3$ and may be due to a lack of vertical resolution. Finally, for most of the cases, when ϵ_{target} is considerably bigger than ϵ_{soil} (contrast > 0.5 approx.), then the bottom reflection is bigger in amplitude than the top one (top/bottom reflection amplitude ratio is < 1). Figure 2.14(b) represents the variation of the number and position of the resonances (amplitude maxima) in function of the permittivity contrast. We determined the resonances automatically from the frequency signatures. The target/soil/contrast conditions are the same as in the previous analysis. In general, we can see that the resonances are growing in number with the contrast and also they shift towards lower frequencies. The collected points for every target describe a set of curves associated to different frequency intervals (orders). The blue and red curves are related to cylinders while the black and the magenta ones to spheres. It is interesting to observe that the diameter of the cylinder does not affect so much the resonances (red and blue curves are overlapping for the different frequencies orders) but for the spheres of 5 and 10cm of diameter, the associated curves are not always coinciding (as happens in the frequency range between 1 and 2 GHz). We can also see that the relative positions of the resonances between the different targets may change depending on the contrast value.

Table 2.2 – Correlation coefficients between time domain signatures related to different target parameters. In red we highlight lower correlation values indicating major changes in the signatures



Height	2cm	3cm	4cm	5cm	6cm	7cm	Depth	3cm	5cm	7cm	10cm	12cm	15cm
2cm	1	0,754	0,771	0,916	0,865	0,888	3cm	1	0,993	0,964	0,954	0,967	0,952
3cm		1	0,731	0,704	0,855	0,791	5cm		1	0,974	0,969	0,982	0,971
4cm			1	0,774	0,831	0,923	7cm			1	0,995	0,971	0,96
5cm				1	0,866	0,854	10cm				1	0,974	0,971
6cm					1	0,908	12cm					1	0,997
7cm						1	15cm						1
Diam.	3cm	5cm	7cm	10cm	13cm	15cm	ε_{target}	3	4	4,5	5,5	6	7
3cm	1	0,98	0,943	0,867	0,776	0,687	3	1	0,957	0,902	0,759	0,697	0,722
5cm		1	0,988	0,941	0,874	0,801	4		1	0,973	0,778	0,646	0,686
7cm			1	0,979	0,933	0,875	4,5			1	0,876	0,876	0,733
10cm				1	0,984	0,946	5,5				1	0,7	0,963
13cm					1	0,987	6					1	0,862
15cm						1	7						1
Rot.	0°	45°X	-45°X	45°Y	90°X	90°Y	ε_{soil}	1	2	4	5	6	7
0°	1	0,6068	0,648	0,677	0,9075	0,9154	1	1	0,951	0,894	0,889	0,886	0,87
45°X		1	0,933	0,9268	0,6358	0,634	2		1	0,941	0,917	0,889	0,86
-45°X			1	0,9651	0,6483	0,6521	4			1	0,992	0,978	0,958
45°Y				1	0,7498	0,7517	5				1	0,993	0,976
90°X					1	0,9953	6					1	0,99
90°Y						1	7						1

With this study we have observed that the time and frequency signatures of landmines are relatively robust to variations in target's depth and soil conditions and very sensitive to varying target permittivity, shape and size due to the internal energy reverberation and the different travelling times through the target. On the other hand, the top/bottom reflection amplitude ratio and the number and position of resonances demonstrated a nearly regular trend with the increasing permittivity contrast. However, for values of ϵ_{target} that are associated to landmines ($\approx$ 3 or less), it is not always possible to distinguish the bottom reflection; similarly, the behavior of the resonances becomes more difficult to interpret, since they are changing quickly in number and position within this range.

Finally, despite the acquired knowledge on the GPR scattering mechanisms for APM recognition, we found out that the features we were able to extract from these synthetic GPR signatures were not

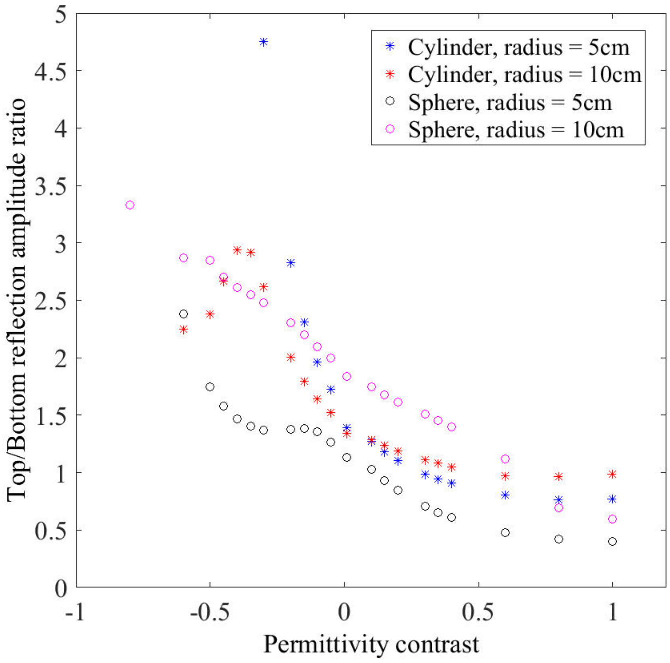


Figure 2.13 – Top/Bottom reflection amplitude ratio behavior of a plastic cylinder buried in a sandy soil ($\epsilon_{soil} = 5$) in function of the permittivity contrast (variable ϵ_{target}.

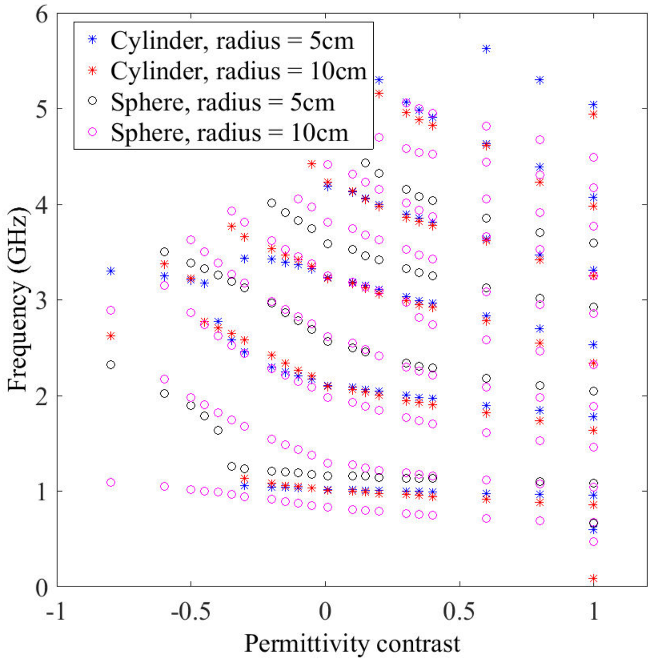


Figure 2.14 – Resonances behavior of a plastic cylinder buried in a sandy soil ($\epsilon_{soil} = 5$) in function of the permittivity contrast (variable ϵ_{target}.

robust enough to be used for experimental data; this motivated further research which finally led to the classification approach proposed in chapter 4. Further studies can implement these descriptors along with other characteristic features. In addition, more complex parametric analysis could also be made introducing heterogeneous or layered soils and rough surfaces in the simulations.

Chapter 3

Sparse Representation and Dictionary Learning techniques

As introduced in section 1.2, given that a signal is sparse in a certain domain, which is usually the case in typical radar scenarios [36], Compressive Sensing (CS) techniques permit the reconstruction of this signal from much less measurements than required by the classical Shannon-Nyquist sampling theorem [15, 66]. This property has a big potential for practical radar applications as one can accelerate the acquisition speed and reduce operational costs.

Sparse representation (SR) and CS share the same framework of techniques but with a slightly different goal. SR tries to exploit the inherent sparsity of data by representing it on an appropriate basis without necessarily reduce the amount of measurements. SR has been succesfully employed in radar for augmented resolution [67, 68], clutter reduction [34] and classification [40, 41, 69–71].

The selection of the dictionary matrix is crucial for obtaining a sparse(r) representation of a signal, i.e., a representation with a minimum number of non-zero coefficients. Depending on the application, one can buld a dictionary with an arbitrary basis (such as Fourier, Wavelets, etc.) [57] or by collecting empirical or synthetic target signatures [40].

Another option is to learn the dictionary from a training set of known data via *dictionary learning* (DL hereafter). In the context of classification, if the training set is *representative* of all the possible signal classes, one can use the sparse non-zero coefficients obtained with the learned dictionary, as a collection of features.

In this chapter we first introduce the basics of SR, describing the properties of the dictionary matrix for correct signal reconstruction. We introduce then some popular and state-of-the-art algorithms for SR, dividing them between greedy and convex optimization approaches. We will then introduce the theoretical fundation of DL and the most important algorithms for learning a dictionary [34]. We will divide these DL approaches into batch-learning (such as K-SVD and LRSDL) [72] and online-learning (such as ODL and CBWLSU) [35, 73]. We will then analyze in detail our novel online dictionary learning approach: the Drop-Off Mini-Batch Online Dictionary Learning (DOMINODL). This is intended to be an improvement over the state-of-the-art online dictionary learning approaches and it will be one of the core contributions of this doctoral thesis aimed at the adaptive classification of GPR signals.

3.1 Fundamentals of Sparse Representation

To introduce the basics of SR, we need to asses the concepts of signal recovery from linear measurements and the process of regularization. We say that the information which are coming from a certain observed scene, are acquired by a measurement process. Assuming that the measurements are real (obtained with linear sensors and then discretized) one can write:

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{3.1}$$

Where $\mathbf{y} \in \mathbb{C}^{M \times 1}$ represents the measurements of the observed scene, arranged in a M dimensional column vector, $\mathbf{D} \in \mathbb{C}^{M \times K}$ represents the model of the scene which can be constructed via a MxK matrix (we will call this the dictionary matrix later) and $\mathbf{x} \in \mathbb{C}^{1 \times K}$ is a coefficient vector which represents the signal information according to the model specified in $\mathbf{D}$. The vector $\mathbf{x}$ represents the information that we want to recover

The problem in 3.1 leads to an equation system which can be: overdetermined (when K<M, there may not be a single solution), determined (when K=M with an unique solution) or underdetermined (when K>M, there will be infinite solution). The latter is the problem of interest for CS and SR.

The solution of an under-determined system of equation is possible if restrictions are applied to $\mathbf{x}$, in this context the concept of regularization comes at hand. Regularization is the process of estimating certain measurement parameters, based on additional a priori information on them. As an example, a regularization problem can be expressed in this way:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \;\; J(\mathbf{x}) \;\; \text{subject to} \;\; \mathbf{y} = \mathbf{D}\mathbf{x} \tag{3.2}$$

Where $J : \mathbb{C} \;\; \rightarrow \;\; \mathbb{C}$ is a cost function, while $\mathbf{y}$ and $\mathbf{x}$ are defined above.A classical cost function used in many regularization problems is $J(\mathbf{x}) = ||\mathbf{x}||_2^2$. One example of regularization process for signal estimation is the matched filter; the regularization problem for this particular application can be expressed like this:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \;\; \frac{|\mathbf{x}^H \mathbf{s}|^2}{E\left\{|\mathbf{x}^H \mathbf{n}|\right\}} \;\; \text{subject to} \;\; \mathbf{y} = \mathbf{s}\mathbf{x} + \mathbf{n}, \tag{3.3}$$

where $\mathbf{s}$ is the transmitted signal, $\mathbf{y}$ is the received one, $\mathbf{n}$ is the noise realization and $\mathbf{x}$ is the parameter that we want to estimate in order to maximize the cost function which, in the case of matched filtering, represent the the signal to noise ratio.

Sparse representation techniques are aimed at the solution of a under-determined linear system as the one indicated in 3.1. The regularization process consists into minimizing a cost function which enforces the sparsity of the vector $\mathbf{x}$ (i.e. the number of its non-zero elements). The problem can be expressed like this:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \;\; ||\mathbf{x}||_0 \;\; \text{subject to} \;\; \mathbf{y} = \mathbf{D}\mathbf{x} \tag{3.4}$$

where $||\mathbf{x}||_0$ is the l_0-norm of the vector $\mathbf{x}$. Formally, the l_p norm of a vector is defined as:

$$||\mathbf{x}||_p = \sqrt[p]{\Sigma_{n=1}^{N}|\mathbf{x}_n|_p}, \;\; 0 < p < \inf \tag{3.5}$$

where N is the number of elements of the vector $\mathbf{x}$. The l_0-norm is a pseudo-norm and can be defined as the limit for $p \to 0$ of $|\mathbf{x}||_p$, resulting in

$$||\mathbf{x}||_0 = card\{n \in [N] : \mathbf{x}_n \neq 0\} \tag{3.6}$$

Equation 3.6 defines the cardinality of the support set of x and this corresponds to the number of its non-zero elements (i.e. what equation 3.4 aims to minimize).
If the observed scene is sparse, then, it can be proven that 3.4 has an unique solution i.e. one can find a matrix $\mathbf{D}$ which gives a perfect reconstruction of the measurements with probability close to one [20]. The number of non-zero elements of the vector $\mathbf{x}$ is also called the *support* of the signal $\mathbf{y}$; if $\mathbf{y}$ is mapped on the vector $\mathbf{x}$ using S non-zero elements then it is said to be *S-sparse*.
The solution of (3.4) is a NP-hard combinatorial problem, however, techniques based on greedy algorithms and convex optimization demonstrated to be efficient in approximating such problem (see 3.2).
We already mentioned the difference between *sensing matrix* and dictionary in chapter 1. Hereafter, we refer to $\mathbf{D}$ as the *dictionary* since the scope of our work is radar target classification, i.e. the matrix $\mathbf{D}$ is not encompassing a phyisical model for a radar signal like in the sensing matrix, rather it contains features from different classes of possible received signals. The columns of $\mathbf{D}$ are called *atoms*. The original signal $\mathbf{y}$ is sparsely represented when it can be espressed with a linear combination of a few atoms of the dictionary $\mathbf{D}$ with the corresponding the non-zero elements of $\mathbf{x}$.
We will assume that matrix $\mathbf{D} \in \mathbb{C}^{M \times K}$ has $M < K$ and that $\mathbf{y}$, the measured signal, is *sparse* or *compressible*. We addressed the concept of sparsity in chapter 1, we say that a signal is compressible when it can be well-approximated by sparse signals, this is an important point since very few real-world signals are *really* sparse. We can quantify the compressibility of a signal $\mathbf{y}$ by calculating the error committed in approximating it like this:

$$\sigma_S(\mathbf{y})_p = \min_{\hat{\mathbf{y}} \in \Sigma_S} \;||\mathbf{y} - \hat{\mathbf{y}}||_p, \tag{3.7}$$

where Σ_S is a set of S-sparse signals, i.e. with $||x||_0 \leq S$.

3.1.0.1 Null space condition, coherence and Restricted Isometric Property (RIP)

There are some conditions that the dictionary matrix $\mathbf{D}$ should fulfill to obtain an exact reconstruction of the original signal, i.e in order that the sparse vector $\mathbf{x}$ will contain the information of the original signal $\mathbf{y}$. We introduce here the null space condition and the Restricted Isometry Property (RIP). We will also explain the concept of coherence for $\mathbf{D}$, an additional property for a succesfull reconstruction of $\mathbf{y}$. The null-space condition is defined by the following equation:

$$\mathcal{N}(\mathbf{D}) = \{\mathbf{z} : \mathbf{D}\mathbf{z} = 0\} \tag{3.8}$$

If we have two vectors $\mathbf{x}, \mathbf{x}' \in \sum_S$ with $\mathbf{Dx} = \mathbf{Dx}'$ then $\mathbf{D}(\mathbf{x} - \mathbf{x}') = 0$ and $\mathbf{x} - \mathbf{x}' \in \sum_{2S}$ represents the null space of $\mathbf{D}$, $\mathcal{N}(\mathbf{D})$ (with $\mathbf{z} = \mathbf{x} - \mathbf{x}'$). This means that $\mathbf{D}$ *uniquely* represents all $\mathbf{x} \in \sum_s$ if and only if $\mathcal{N}(\mathbf{D})$ contains no vectors in $\sum_{2S}$. In other words, $\mathbf{D}$ obeys the null space condition when two distinct vectors $\mathbf{x}, \mathbf{x}' \in \sum_S$ will generate two different measurement reconstructions $\mathbf{Dx} \neq \mathbf{Dx}'$ or, conversely, any single measurement $\mathbf{y}$ will accept an unique sparse representation $\mathbf{x} \in \sum_S$ in $\mathbf{D}$.
The coherence of the dictionary matrix $\mathbf{D}$ is also an important factor for satisfactory recovery. Let $\mathbf{D} \in \mathbb{C}^{MxK}$ being the dictionary matrix with column vectors $\mathbf{d_k}$ and $||\mathbf{d_k}||_2 = 1$. The coherence of $\mathbf{D}$ is defined as:

$$\mu(\mathbf{D}) := \max_{1 \leq i < j \leq K} \frac{| < \mathbf{d_i}, \mathbf{d_j} > |}{||\mathbf{d_i}||_2 ||\mathbf{d_j}||_2} : \; i, j \in \mathbb{N} \tag{3.9}$$

$\mu(\mathbf{D})$ values are between 0 and 1; if $\mathbf{D}$ is a quadratic matrix and all the columns of $\mathbf{D}$ are orthonormal between each other then $\mu(\mathbf{D}) = 0$, conversely when all columns are completely correlated we have $\mu(\mathbf{D}) = 1$. For under-determined systems, where $M < K$, which are of interest for SR, a small coherence means that all sub-columns matrices $\mathbf{D_j}$ with $|j| \leq M$ are well-conditioned thus, for SR, the coherence should be the smallest possible.
Respect to the null-space propriety, the Restricted Isometry Property (RIP) takes in account the noise in the measurements for the correct reconstruction [15, 74]. RIP is used to indicate if the sparse vectors $\mathbf{x}$ are actually related to measurements $\mathbf{y}$ which have energy close to $\mathbf{x}$. We say that $\mathbf{D}$ possess the RIP of order S (being S the support of $\mathbf{y}$ in $\mathbf{D}$) if exist a $\delta_S(\mathbf{D})$ such that the following equation:

$$(1 - \delta_S)||\mathbf{x}||_2^2 \;\leq\; ||\mathbf{Dx}||_2^2 \;\leq\; (1 + \delta_S)||\mathbf{x}||_2^2, \tag{3.10}$$

holds for all $\mathbf{x} \in \Sigma_S$. If $\mathbf{D}$ satisfies the RIP property of order $2S$, given by $\delta_{2S} \leq \delta \approx \sqrt{2} - 1$, then with high probability all S-sparse vectors are successfully recovered [15, 74].

3.2 Sparse Representation methods

As mentioned in the previous section, the problem presented in (3.4) is a NP-hard problem and it is in general not tractable, in this section we will describe some typical techniques to approximate its solution dividing them into greedy and convex optimization approaches. Popular greedy approaches to solve 3.4 (such as matching pursuit (MP), orthogonal matching pursuit (OMP), block orthogonal matching pursuit (BOMP), etc. [27, 28]) approximate the l_0-norm solution in 3.4 using iterative strategies; they are computationally fast but may not lead to a global optimum solution. State of the art algorithms for sparse representation such as basis pursuit (BP), basis pursuit de-noising (BPDN) [30, 31], LASSO [75] and LARS [76] aim to approximate (3.4) by using less restrictive constraints (norm-1 and norm-2) which can still assure a sparse solution for $\mathbf{x}$ [29]. These methods are based on convex optimization, they are more robust against noise and therefore they may lead to a better reconstruction of $\mathbf{y}$ but they are computationally more demanding. In this section we will describe in detail the SR techniques that we employed the most in our work: the Orthogonal matching pursiut (OMP) (along with a faster implementation of it) along with its batch version (batch-OMP) [72]. Convex optimization methods (such as BPDN, LASSO and LARS) will also be introduced since they have been used for the dictionary update step of some of the proposed DL algorithms (see 3.4).

3.2.1 Greedy approaches

Greedy strategies for SR are iterative approaches which searches for the best local optimal solution for each iteration (i.e. the *greediest* one) hoping that it coincides to the global best one. They can approximate the solution of the problem in 3.4. These algorithms chooses only the most appropriate elements in $\mathbf{x}$ according to the input constraint (which is the expected sparsity or the reconstruction error, as we will explain right away) to reconstruct the measurement vector $\mathbf{y}$ [77]. We start by describing in detail the Orthogonal Matching Pursuit (OMP), a variation of the Matching Pursuit (MP) algorithm.

3.2.1.1 Orthogonal matching pursuit (OMP)

The OMP problem has a dual formulation:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \ ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \ \text{ subject to } \ ||\mathbf{x}||_0 \leq S \tag{3.11}$$

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \ ||\mathbf{x}||_0 \ \text{ subject to } \ ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \leq \delta \tag{3.12}$$

Equation (3.11) constraints the sparsity number S while (3.12) constraints the l_2-norm of the residual by the parameter δ. OMP is an iterative process, at each iteration it search for an atom $\mathbf{d_i}$ wich gives the best projection onto the residual signal $\mathbf{y} - \mathbf{Dx}$ and obtains the value of the corresponding coefficient $\mathbf{x_i}$ by least square. Our implementation of OMP can use either the sparsity number S or the residual error δ as a stopping criterion, algorithm 1 shows the details of its implementation which is based on the work in [27].

Algorithm 1: Orthogonal Matching Pursuit (OMP)

```
Input: Measurement vector ($\mathbf{y} \in \mathbb{C}^{Mx1}$), Dictionary ($\mathbf{D} \in \mathbb{C}^{MxK}$), Sparsity number ($S$) or Residual error threshold ($\delta$)
Output: Sparse coefficients vector ($\mathbf{x} \in \mathcal{R}^{1xK}$)
Initialize iteration count $i = 1$
Initialize the residual as the measurement vector: $\mathbf{r_i} = \mathbf{y}$
Initialize the vector of sparse coefficients: $\mathbf{x} = \underline{\mathbf{0}} \in \mathcal{C}^{1xn}$
Initialize the matrix of new found atoms: $\mathcal{D}_0 = \emptyset$
Loop until $i = S$ or $||\mathbf{r_i} - \mathbf{Dx}||_\mathbf{2} \geq \sigma$
    Find the atom $\mathbf{d_i}$ which gives the maximum dot product with $\mathbf{r}$: $\mathbf{di} = \underset{\mathbf{d_i}}{\operatorname{argmax}} \langle \mathbf{r_i}, \mathbf{d_i} \rangle$
    Normalize the new found atom by the norm-2 : $\mathbf{d_i} = \frac{\mathbf{d_i}}{||\mathbf{d_i}||_2}$
    Include $\mathbf{d_i}$ in the set of new found atoms: $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \mathbf{d_i}$
    Update $x_i$ by solving a least square problem: $\mathbf{x_i} = \underset{\mathbf{x}}{\operatorname{argmin}}||\mathbf{y} - \mathcal{D}_i\mathbf{x}||_2$
    Update residual: $\mathbf{r_i} = \mathbf{r_{i-1}} - \mathcal{D}_i\mathbf{x_i}$
EndLoop
```

We initialize the residual vector $\mathbf{r_i}$ as the measurement vector $\mathbf{y}$ and we create an empty set $\mathcal{D}$ which will contain the atoms of $\mathbf{D}$ which will be used for the sparse decomposition of $\mathbf{y}$ and updated during the iterations. If the input $S \neq 0$ and $S \in \mathbb{N}$, the maximum iterations count correspond to the desired sparsity number. In many radar applications, this is equivalent to select the number of expected targets in the scene which we want to detect, in other words, we are making an assumption on the support of the vector $\mathbf{y}$. If $S = 0$ and/or $\delta \in \mathbb{R}$ with $0 \leq \delta \geq 1$ we are ignoring the sparsity number and the stopping criterion depends on the error: $||\mathbf{r_i} - \mathbf{Dx}||_2$. For every iteration, the algorithm finds $\mathbf{d_i}$, the atom of $\mathbf{D}$ which has the bigger dot product with the residual $\mathbf{r_i}$. The new found atom is normalized and included to the set $\mathcal{D}$ at the current iteration ($\mathcal{D}_i$). The coefficient update step is what differentiate OMP respect to MP and it is obtained by a least square regression problem (see step 9) [27]. Finally, the residual error is updated before evaluating the stopping criterion and evenually go on with the iterations.

3.2.1.2 A faster implementation of OMP

The least square problem at step 9 of algorithm 1 is computationally demanding and it is usually computed by using a Cholesky factorization or a QR decomposition [72]. To speed up computations we propose a faster variant of OMP (see 2). The difference in this implementation is that the new found atom is orthogonalized against all the previously found one $\mathbf{d_j}$ with $j = 1, \cdots, i-1$ via Gram–Schmidt process (see step 10) [78] before the coefficient update. Having an orthonormal basis among the new found atoms is beneficial for the coefficient update where, instead of solving a least square problem, we can directly obtain the updated coefficients $\mathbf{x}$ by projecting them over the residual $\mathbf{r_i}$ (see step 13). After the stopping criterion is met, we adjust the coefficient vector obtained at the last iteration ($\mathbf{x_i}$) solving just one least square problem without orthogonalizing before, this make possible that the coefficients $\mathbf{x}$ have the right coefficient values respect to $\mathbf{y}$.

Algorithm 2: A faster implementation of OMP

```
Input: Measurement vector ($\mathbf{y} \in \mathbb{C}^{mx1}$), Dictionary ($\mathbf{D} \in \mathbb{C}^{mxn}$), Sparsity number ($S$) or Residual error threshold ($\delta$)
Output: Sparse coefficients vector ($\mathbf{x} \in \mathbb{C}^{1xn}$)
1  Initialize iteration count $i = 1$
2  Initialize the residual as the measurement vector: $\mathbf{r_i} = \mathbf{y}$
3  Initialize the vector of sparse coefficients: $\mathbf{x} = \underline{\mathbf{0}} \in \mathbb{C}^{1xn}$
4  Initialize the matrix of new found atoms: $\mathcal{D}_0 = \emptyset$
5  Loop until $i = S$ or $||\mathbf{r_i} - \mathbf{Dx}||_2 \geq \delta$
6      Find the atom $\mathbf{d_i}$ which gives the maximum dot product with $\mathbf{r}$: $\mathbf{di} = \underset{\mathbf{d_i}}{\text{argmax}} \langle \mathbf{r_i}, \mathbf{d_i} \rangle$
7      Normalize the new found atom by the norm-2 : $\mathbf{d_i} = \frac{\mathbf{d_i}}{||\mathbf{d_i}||_2}$
8      Include $\mathbf{d_i}$ in the set of new found atoms: $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \mathbf{d_i}$
9      for $j = 1, \cdots, i-1$
10         Orthogonalize $\mathbf{d_i}$ against all previously found $\mathbf{d_j}$: $\mathbf{d_i} = \mathbf{d_i} - \langle \mathbf{d_j}, \mathbf{d_i} \rangle \mathbf{d_j}$
11     end for
12     for $j = 1, \cdots, i$
13         Coefficients update $\mathbf{x_i} = \langle \mathbf{x_j}, \mathbf{r_j} \rangle$
14     end for
15 EndLoop
16 UPDATE $x_i$ by solving a least square problem: $\mathbf{x_i} = \underset{\mathbf{x}}{\text{argmin}}||\mathbf{y} - \mathcal{D}_i\mathbf{x}||_2$
17 Update residual: $\mathbf{r_i} = \mathbf{r_{i-1}} - \mathcal{D}_i\mathbf{x_i}$
```

3.2.1.3 Batch-OMP

Batch-OMP is a variant of the classical OMP and it is used when a large number of signals must be represented over the same dictionary. The intuition behind this method is that it is not required to compute the residual $\mathbf{r}$ at each iteration for the atom selection step, only $\mathbf{D}^T\mathbf{r}$ is required. The atom selection step can be rewritten in order to exploit its dependance only on the dictionary at the

iteration i (called $\mathbf{D_i}$) and the measurement vector $\mathbf{y}$ [79] without calculating $\mathbf{D}^T\mathbf{r}$.
Calling $\boldsymbol{\alpha} = \mathbf{D}^T\mathbf{r}$, $\boldsymbol{\alpha}^0 = \mathbf{D}^T\mathbf{y}$ and $\mathbf{G} = \mathbf{D}^T\mathbf{D}$ one can write;

$$\begin{aligned}\boldsymbol{\alpha} = \mathbf{D}^T(\mathbf{y} - \mathbf{D}_I(\mathbf{D}_I)^+\mathbf{y}) \\ = \boldsymbol{\alpha}^0 - \mathbf{G}_I(\mathbf{D}_I)^+\mathbf{y} \\ = \boldsymbol{\alpha}^0 - \mathbf{G}_I(\mathbf{D}_I^T\mathbf{D}_I)^{-1}\mathbf{D}_I^T\mathbf{y} \\ = \boldsymbol{\alpha}^0 - \mathbf{G}_I(\mathbf{G}_{I,I})^{-1}\boldsymbol{\alpha}_I^0\end{aligned} \tag{3.13}$$

Given the pre-computed $\boldsymbol{\alpha}^0$ and $\mathbf{G}$, one can compute $\boldsymbol{\alpha}$ without explicitly computing $\mathbf{r}$. However, if the residual is not computed, it is impossible to set a stopping criterion based on the error. In [72] an extention of batch-OMP where the residual l_2 norm $||\mathbf{r}||_2^2$ at the iteration i is calculated by an incremental formula; this permits the use of the residual as a stopping criterion without explicitly computing it; see [72] for the implementation details.
Batch-OMP will be used in some of the proposed DL algorithms (see chapter 4) and in presented, novel algorithm for online dictionary learning (DOMINODL).

3.2.2 Convex optimization approaches

A convex optimization problem deals with miminizing convex functions over a convex set; the convexity makes the minimization easier since the local minimum of the fuction is also the global one [80]. Convex optimization methods for SR, contrary to greedy approaches, aim to minimize the l_1-norm of the coefficient vector x.

$$\hat{\mathbf{x}} = \underset{x}{\text{argmin}} \ ||\mathbf{x}||_1 \text{ subject to } \mathbf{D}\mathbf{x} = \mathbf{y} \tag{3.14}$$

The objective function $||.||_1$ is in fact a convex and tractable, whereas $||.||_0$ is non-convex and generally very difficult to solve. The problem in 3.14 is known as basis pursuit (BP); it can be solved efficiently using linear programming techniques and it is demonstrated [20] that it still leads to a sparse approximation of the underdetermined system solution in 3.4.
In case we want to take in account for the noise in the data, we can make an additional approximation:

$$\hat{\mathbf{x}} = \underset{x}{\text{argmin}} \ ||\mathbf{x}||_1 \text{ subject to } ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \ \leq \delta \tag{3.15}$$

In this way we don't want to exactly solve 3.14 but we make an approximation based on a positive parameter δ which amplitude (between 0 an 1 if the data is normalized) corresponds to the noise level, this parameter plays the same role of δ in OMP (see 2). This problem is called basis pursuit denoise (BPDN).
There are different derivation that we can extract from 3.15 like the LASSO problem [75]

$$\hat{\mathbf{x}} = \underset{x}{\text{argmin}} \ ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \text{ subject to } ||\mathbf{x}||_1 \ \leq \tau \tag{3.16}$$

and the penalized least square problem

$$\hat{\mathbf{x}} = \underset{x}{\text{argmin}} \ ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2 \ + \lambda||\mathbf{x}||_1 \tag{3.17}$$

Where λ is a regularization parameter [81]. For an appropriate choice of δ, τ and λ the solutions of these three approaches coincide.
One of the most popular algorithm for solving the problems in 3.15 and 3.16 was proposed from Van Der Berg and Friedlander and it is based on the relation between LASSO and BPDN, defined by the Pareto curve. The Pareto curve indicates the optimal trade off between two-norm of the residual ($||\mathbf{y} - \mathbf{D}\mathbf{x}||_2$) and the one-norm of the solution $\mathbf{x}$. The solution of a BPDN problem using this approach consists on solving a sequence of LASSO problem using spectral projected gradient and use the Newton method applied to the Pareto curve to observe how much the solution of 3.16 get close to 3.15; the process stops when satisfactory threshold is met [30, 31].
Homotopy approaches [82] such as LARS [76] solve the BPDN problem by repeatedly solving 3.17 for all possible values of λ. LARS is an algorithm for both variable and coefficient selection which is aimed to fit a linear regression model to high dimensional data (in SR we can identify the atoms of the dictionary as variables and the sparse elements of the vector $\mathbf{x}$ as coefficients). LARS has been already proposed for l_1 minimization problems [83] and it is convenient to use it to solve the penalized least square problem in 3.17 because of its speed (comparable to forward selection algorithms) and for the fact that it produces a full piecewise linear solution path (i.e. for every λ) [84]. Moreover the LASSO problem can be derived by LARS, in fact, LASSO can be seen as a variation of a ridge-regression problem [84] where the focus is on coefficients selection rather to variable estimation but with the constraint of putting some coefficients to zero. In a nutshell, refering to equation 3.16, the basic steps of LARS are:

- Start with all coefficients $\mathbf{x}_j$ with ($j = 1...K$) equal to zero, set the initial residual $\mathbf{r}$ as $\mathbf{y}$
- Find the atom $\mathbf{d}_j$ which has the highest correlation with the $\mathbf{r}$
- Increase the coeficient $\mathbf{x}_j$ in the direction of the sign of its correlation with $\mathbf{y}$, calculate the residual $\mathbf{r} = ||\mathbf{y} - \mathbf{D}\mathbf{x}||_2^2$ for each increment and stop when some other atom $\mathbf{d}_k$ has as as much correlation with $\mathbf{r}$ as $\mathbf{d}_j$ has.
- Increase $\mathbf{x}_j, \mathbf{x}_k$ in the direction defined by their joint least-square coefficient of the current residual on $\mathbf{d}_j, \mathbf{d}_k$, until some other $\mathbf{d}_m$ has as much correlation with the residual $\mathbf{r}$
- Continue until all atoms are considered in the model

The LARS-LASSO problem is proposed in [76] and will be used later in this work for the sparse decomposition step of the Online Dictionary Learning (ODL) algorithm.

3.3 Fundamentals of Dictionary Learning

The dictionary $\mathbf{D}$ may be learned from the data it is going to represent. *Dictionary learning* (DL) techniques aim to create adapted dictionaries which provide the sparsest reconstruction for given training-sets, i.e., a representation with a minimum number of constituting atoms. DL methods are critical building blocks in many applications such as deep learning, image denoising, and super-resolution; see [85, 86].
Please note that, at the time of writing this dissertation, the DL algorithms described in this section

could only be applied on real data (whereas the SR techniques presented previously could also work on complex data).

The first step in DL consists of building a training database $\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1 & \cdots & \mathbf{y}_L\end{bmatrix} \in \mathbb{R}^{MxL}$ of L vectors, each with M elements. We assume that every vector $\mathbf{y}_i$ is generated by a linear combination of the K atoms of a certain dictionary matrix $\mathbf{D} \in \mathbb{R}^{MxK}$ and an associated sparse vector of coefficients $\mathbf{x_i}$, having $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \cdots & \mathbf{x}_L\end{bmatrix} \in \mathbb{R}^{KxL}$. The core problem of DL is to find the dictionary $\mathbf{D}$ which give the set of sparsest solution $\mathbf{X}$, whe have then:

$$\mathbf{Y}_{(MxL)} = \mathbf{D}_{(MxK)}\ \mathbf{X}_{(KxL)}, \tag{3.18}$$

where the dimension L is usually larger than the number of the dictionary elements ($L > K$). The DL problem has a dual formulation, whether it constraints sparsity (3.19) or the model deviation (3.20):

$$\begin{aligned}\hat{\mathbf{D}}, \hat{\mathbf{X}} = \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}}\ \ \|\mathbf{Y} - \mathbf{DX}\|_F \\ \text{subject to}\ \ \|\mathbf{x}_i\|_0 \leq S,\ 1 \leq i \leq L.\end{aligned} \tag{3.19}$$

$$\begin{aligned}\hat{\mathbf{D}}, \hat{\mathbf{X}} = \underset{\mathbf{D},\mathbf{X}}{\operatorname{argmin}}\ \ \|\mathbf{Y} - \mathbf{DX}\|_F \\ \text{subject to}\ \ \|\mathbf{y}_i - \mathbf{Dx}_i\|_2 \leq \epsilon,\ 1 \leq i \leq L.\end{aligned} \tag{3.20}$$

Where S is the sparsity number, i.e. the desired number of the non-zero elements in the decomposed vector and ϵ is the residual error.

Since both $\mathbf{D}$ and $\mathbf{x}$ are unknown, a common approach is to use alternating minimization in which we start with an initial guess of $\mathbf{D}$ and then obtain the solution iteratively by alternating between two stages: *sparse representation* and *dictionary update* [87] - as follows:

1) Sparse representation: Obtain $\mathbf{X}_{(t)}$ for each $\mathbf{y}_i$ as:

$$\begin{aligned}\hat{\mathbf{X}}_{(t)} = \underset{\mathbf{X}}{\operatorname{argmin}}\ \ \left\|\mathbf{Y} - \mathbf{D}_{(t-1)}\mathbf{X}\right\|_F \\ \text{subject to}\ \ \left\|\mathbf{x}_{i_{(t-1)}}\right\|_p \leq S,\ 1 \leq i \leq L,\end{aligned} \tag{3.21}$$

where $\mathbf{X}_{(t)}$ is the SR in t^{th} iteration. This can be solved using greedy algorithms such as orthogonal matching pursuit (OMP) ($p = 0$) or convex relaxation methods like basis pursuit denoising (BPDN) ($p = 1$).

2) Dictionary Update: Given $\mathbf{X}_{(t)}$, update $\mathbf{D}_{(t)}$ such that

$$\mathbf{D}_{(t)} = \underset{\mathbf{D}\in\mathcal{D}}{\operatorname{argmin}}\ \ \left\|\mathbf{Y} - \mathbf{DX}_{(t)}\right\|_F, \tag{3.22}$$

where $\mathcal{D}$ is a set of all dictionaries with unit column-norms, $\left\|\mathbf{d}_j\right\|_2 = 1$ for $1 \leq j \leq K$. This subproblem is solved by methods such as singular value decomposition or gradient descent [35, 87].

3.4 Dictionary Learning methods

In this section we will analyze some state-of-the-art DL algorithms that we used for the classification approach described in section 4, dividing them in batch-DL and online-DL approaches. Regarding batch-DL, we will focus on the description of the K-SVD, one of the most popular DL algorithms [72] and the more recent LRSDL [88] due to its capability of generating a class-discriminative dictionary. Regarding online-DL techniques, we will analyze the state-of-the-art method ODL [35], the recent CBWLSU [35, 73] and describe the development of a novel Online-DL strategy named Drop-Off Mini-batch Online Dictionary Learning (DOMINODL) [42].

3.4.1 Batch DL approaches

Classical methods for DL such as Method of Optimal Directions (MOD) [33] and K-SVD [34] retain a guess for $\mathbf{D}$ and $\mathbf{X}$ and iteratively update either $\mathbf{X}$ using basis/matched pursuit or $\mathbf{D}$ using least squares or singular value decomposition (SVD). Batch-DL methods deal with the entire training set for every iteration, although very successful, they are computationally demanding and not scalable to high-dimensional training sets.

Discriminative DL algorithms such as Label Consistent K-SVD [89] and discriminative K-SVD [90] still classify as batch DL techniques (since they deal with the entire training set) but they learn a class-discriminative dictionary which is separated in subsets of atoms belonging to different classes present in the training set. Low Rank Shared Dictionary Learning (LRSDL [88]) is an interesting discriminative DL technique which adress some shortcomings its precursors and has therefore been tested for our classification strategy.

3.4.1.1 K-SVD

K-SVD iteratively alternate a sparse decomposition and a dictionary update step solving a minimization over the number of non-zero elements in the set of representation vectors $\mathbf{X}$, and one over $\mathbf{D}$ for updating the dictionary.

For the sparse coding step at the iteration t, K-SVD solves OMP for each element $\mathbf{y_i}$ in the training set $\mathbf{Y}$:

$$\begin{aligned} \hat{\mathbf{x}}_i &= \underset{\mathbf{x}_i}{\operatorname{argmin}} \quad \|\mathbf{x}_i\|_0 \\ \text{subject to} & \quad \left\|\mathbf{y}_i - \mathbf{D}_{(\mathbf{t-1})}\mathbf{x}_i\right\|_2^2 \leq \delta,\ \forall 1 \leq i \leq L, \end{aligned} \tag{3.23}$$

where i is the index which represent the training set elements, $\mathbf{D}_{(\mathbf{t-1})}$ is the dictionary computed at the previous iteration $(t-1)$ and δ is the maximum residual error.

K-SVD shares the same strategy for the sparse decomposition step with MOD, however, the dictionary update rule is different. Let K be an input parameter of K-SVD which indicates the number of columns of the learned dictionary. For the dictionary update step, K-SVD solves the global minimization problem in (3.22) via K sequential minimization problems, wherein every column $\mathbf{d}_k$ of $\mathbf{D}$ and its corresponding row of coefficients $\mathbf{X}_{\text{row},k}$ of $\mathbf{X}$ are updated. The dictionary update step for K-SVD is explained in algorithm 3.

Let's assume that we are at the iteration t and we already peformed the sparse decomposition step using the dictionary which has been obtained in the iteration $t-1$. For each kth dictionary atom, we calculate the error term $\mathbf{Y}_r = \mathbf{Y} - \sum_{l \neq k} \mathbf{d}_{l_{t-1}} \mathbf{X}_{\text{row},l_{(t-1)}}$ and extract a subset $\mathcal{Y}_k$ of it which comprises the elements of $\mathbf{Y_r}$ which use the selected atom (k). Then we use SVD to find the closest rank-1 approximation of $\mathcal{Y}_k$ to obtain $\mathbf{d}_k$ subjected to the constraint $\|\mathbf{d}_{k_{(t)}}\|_2 = 1$.

K-SVD global optimization will terminate after a series of sparse decomposition plus dictionary update steps depending on the changes on $\left\|\mathbf{Y} - \mathbf{D}\mathbf{X}_{(\mathbf{t})}\right\|_F^2$.

Algorithm 3: K-SVD dictionary update step

```
Input: $\mathbf{D}_{(t-1)}$, $\mathbf{X}_{(t)}$ and residual error threshold ($\delta$)
Output: $\mathbf{D}_{(t)}$
Initialize atom index: $k = 1$
Loop until $k = K$
    Select $\mathbf{d}_{k_{(t-1)}}$ and the corresponding $\mathbf{X}_{\text{row},k_{(t)}}$
    Compute $\mathbf{Y}_r = \mathbf{Y} - \sum_{l \neq k} \mathbf{d}_{l_{(t-1)}} \mathbf{X}_{\text{row},l_{(t-1)}}$
    SR of $Y_r$ using OMP with residual error threshold = $\delta$
    Select a subset of signals $\mathcal{Y}_k$ from $\mathbf{Y}_r$ which use the atom $\mathbf{d}_{k_{(t-1)}}$ in their SR.
    Compute SVD of $\mathcal{Y}_k$ for obtaining both $\mathbf{d}_{k_{(t)}}$ and $\mathbf{X}_{\text{row},k_{(t)}}$
EndLoop
```

For this work, a particular K-SVD implementation was used [72]. This version employs a faster approximation for the SVD step in the dictionary update and uses batch-OMP for the sparse decomposition step, making it more feasible to deal with large sets of signals.

The performance of K-SVD can be improved in terms of both computational complexity and obtaining an incoherent dictionary if the learning process enforces constraints such as hierarchical tree sparsity [91], structured group sparsity (StructDL) [92], Fisher discrimination (FDDL) [93], and low-rank-and-Fisher (D^{2}L^{2}R^2) [94].

3.4.1.2 LRSDL

LRSDL [88] is one of the latest evolution of discriminative DL algorithms; we tested it on our classification approach due to its promising capabilities for class recognition. In this subsection we try to give the basic idea of discriminative-DL and briefly describe the theoretical background on which LRSDL is based of.

Discriminative DL algorithms, as D-KSVD [90] and LC-KSVD [89], employ a learning strategy which promotes the generation of a dictionary $\mathbf{D}$ which is separated in blocks of atoms associated to different classes as $\mathbf{D} = [\mathbf{D}_1, \cdots, \mathbf{D}_C] \in \mathbb{R}^{M \times K}$ where C is the number of classes present in the training set $\mathbf{Y}$. The resultant coefficient matrix $\mathbf{X}$ is close to be sparse with all non-zeros being one while satisfying a block diagonal structure.

However, objects belonging to different classes often have common features, therefore the assumption of non-overlapping subspaces done by such algorithms is often unrealistic in practice. Techniques such as DL with structured incoherence and shared features (DLSI) [95], separating the commonality and the particularity (COPAR) [96] and convolutional sparse DL (CSDL) [97] exploit common patterns among different classes even though different objects possess distinct class-specific features. These methods produce am additional constituent $\mathbf{D_0}$ which is shared among all classes so that $\mathbf{D} = [\mathbf{D}_1, \cdots, \mathbf{D}_C, \mathbf{D}_0] \in \mathbb{R}^{M \times K}$. The drawback of these strategies is that the shared dictionary may also contain class-discriminative features.

To avoid this problem, LRSDL [88] requires that the shared dictionary must have a low-rank structure and that its sparse coefficients have to be almost similar. LRSDL learns both $\mathbf{D}$ and $\mathbf{X}$ by solving a minimization problem with a cost function which is closely related to the one of another DL algorithm called Fisher Discriminative Dictionary Learning (FDDL) [98]. FDDL tries to minimize the discriminative fidelity $\mathbf{J_y}(\mathbf{D},\mathbf{X})$, defined as:

$$\mathbf{J_y}(\mathbf{D},\mathbf{X}) = \frac{1}{2}\mathbf{f_y}(\mathbf{D},\mathbf{X}) + \lambda_1||\mathbf{X}||_1 + \frac{\lambda_2}{2}g(\mathbf{X}) \tag{3.24}$$

with $\mathbf{Y} \in \mathbb{R}^{M\times L}$ being the class-labeled training set with with $c = 1...C$ classes and $L = \sum_{c=1}^{C} n_c$ being its number of elements with n_c being the number of elements for the class c. $\mathbf{D} = [\mathbf{D}_1, \cdots, \mathbf{D}_C] \in \mathbb{R}^{M\times K}$ being the dictionary with $K = \sum_{c=1}^{C} k_c$ with k_c being the number of atoms for the class c, λ_1 and λ_2 being two fidelity terms and $\mathbf{f_y}(\mathbf{D},\mathbf{X})$ being:

$$\mathbf{f_Y}(\mathbf{D},\mathbf{X}) = \sum_{c=1}^{C} \mathbf{r}_{\mathbf{Y}_c}(\mathbf{D},\mathbf{X}_c) \tag{3.25}$$

with $\mathbf{Y}_c \in \mathbb{R}^{M\times n_c}$ being the training set elements belonging to the class c and $\mathbf{X}_c \in \mathbb{R}^{k_c\times n_c}$ being its SR and $\mathbf{r}_{\mathbf{Y}_c}(\mathbf{D},\mathbf{X}_c)$ being:

$$\mathbf{r}_{\mathbf{Y}_c}(\mathbf{D},\mathbf{X}_c) = ||\mathbf{Y}_c - \mathbf{D}\mathbf{X}_c||_F^2 + ||\mathbf{Y}_c - \mathbf{D}_c\mathbf{X}_c^c||_F^2 + \sum_{j\neq c}||\mathbf{D}_j\mathbf{X}_c^j||_F^2 \tag{3.26}$$

with $\mathbf{X}_c^c \in \mathbb{R}^{k\times n_c}$ being the sparse coefficients of $\mathbf{Y}_c$ on $\mathbf{D}_c$ and the last term ensures that the minimization process leads to a $\mathbf{D}_j$ with a small contribution to the representation of $\mathbf{Y}_c$ for all $j \neq c$. $\mathbf{g}(\mathbf{X})$ in 3.25 is called discriminant constraint and it is expressed as:

$$\mathbf{g}(\mathbf{X}) = \sum_{c=1}^{C}(||\mathbf{X}_c - \mathbf{M}_c||_F^2 - ||\mathbf{M}_c - \mathbf{M}||_F^2) + ||\mathbf{X}||_F^2 \tag{3.27}$$

where $\mathbf{M}_c = \mu(\mathbf{X}_c) \in \mathbb{R}^{k_c\times n_c}$ and $\mathbf{M} = \mu(\mathbf{X}) \in \mathbb{R}^{k\times L}$ with $\mu(\mathbf{X})$ being an operator which calculates the mean vectors of $\mathbf{X}$ and stack them column-wise as many times as the columns of $\mathbf{X}$ to generate a matrix $\mathbf{M}$. The last term in 3.27 make sure that the cost function 3.24 became convex w.r.t. $\mathbf{X}$. The number of columns in the first two terms depends on context, e.g. by writing $\mathbf{M}_c - \mathbf{M}$, we mean that $n = nc$.

We are now including the shared dictionary $\mathbf{D_0}$ in the presented FDDL formulation.

Let $\bar{\mathbf{D}} = [\mathbf{D}_1, \cdots, \mathbf{D}_C, \mathbf{D}_0] \in \mathbb{R}^{M\times K}$ be the total dictionary with $\bar{\mathbf{X}} = [\mathbf{X}^T, (\mathbf{X}^0)^T]$ and $\bar{\mathbf{X}}_c = [\mathbf{X}_c^T, (\mathbf{X}_c^0)^T]$. The discriminative fidelity term $\mathbf{f_Y}(\mathbf{D},\mathbf{X})$ becomes $\mathbf{f}_{\bar{\mathbf{Y}}}(\bar{\mathbf{D}},\bar{\mathbf{X}}) = \sum_{c=1}^{C}\bar{\mathbf{r}}_{\mathbf{Y_c}}(\bar{\mathbf{D}},\bar{\mathbf{X}}_\mathbf{c})$ with $\bar{\mathbf{r}}_{\mathbf{Y_c}}(\bar{\mathbf{D}},\bar{\mathbf{X}}_\mathbf{c})$ defined as:

$$\bar{\mathbf{r}}_{\mathbf{Y_c}}(\bar{\mathbf{D}},\bar{\mathbf{X}}_c) = ||\mathbf{Y}_c - \bar{\mathbf{D}}\bar{\mathbf{X}}_c||_F^2 + ||\mathbf{Y}_c - \mathbf{D}_c\mathbf{X}_c^c - \mathbf{D}_0\mathbf{X}_c^0||_F^2 + \sum_{j\neq c}||\mathbf{D}_j\mathbf{X}_c^j||_F^2 \tag{3.28}$$

Since $\bar{\mathbf{r}}_{\mathbf{Y_c}}(\bar{\mathbf{D}},\bar{\mathbf{X}}_\mathbf{c}) = \mathbf{r}_{\bar{\mathbf{Y}}_\mathbf{c}}(\mathbf{D},\mathbf{X_c})$ with $\bar{\mathbf{Y}}_c = \mathbf{Y}_c - \mathbf{D}_0\mathbf{X}_c^0$, we have $\bar{\mathbf{f}}_{\mathbf{Y_c}}(\bar{\mathbf{D}},\bar{\mathbf{X}}_\mathbf{c}) = \mathbf{f}_{\bar{\mathbf{Y}}_\mathbf{c}}(\mathbf{D},\mathbf{X_c})$ with $\bar{\mathbf{Y}} = \mathbf{Y} - \mathbf{D}_0\mathbf{X}^0$.

The aforementioned expressions indicate that $\mathbf{Y}_c$ must be represented by a combination of $\mathbf{D}_c$ and $\mathbf{D}_0$ and therefore the shared atoms of $\mathbf{D}_0$ must represent samples from all c classes

The other assumption which LRSDL does (unlike other discriminative approach which use a shared dictionary such as COPAR [96]) is that $\mathbf{D}_0$ must have low rank to prevent that it will include class-discriminative atoms. LRSDL use a regularization on the nuclear norm $||\mathbf{D}_0||_*$ which is the convex relaxation of the $rank(\mathbf{D}_0)$ to force the shared dictionary in having low rank.

Finally, LRSDL levels the contributions of the shared dictionary $\mathbf{D}_0$ to all the C classes in the training set. The regularization term $||\mathbf{X}^0 - \mathbf{M}^0||_F^2$ is added to the objective function $\mathbf{g}(x)$ in 3.27 to force each $\mathbf{X}^0$ to be close to the mean vector of all $\mathbf{X}^0$.

$$\mathbf{g}(\bar{\mathbf{X}}) = \mathbf{g}(\mathbf{X}) + ||\mathbf{X}^0 - \mathbf{M}^0||_F^2 \tag{3.29}$$

The LRSDL dictionary update step employs alternating direction method of multipliers (ADMM) [99] and FISTA [100] for the sparse decomposition step. Once the data is sparsely represented with such dictionaries, a sparse-representation-based classifier (SRC) is used to predict the class of new data.

3.4.2 Online DL approaches

Batch-Learning approaches, like K-SVD, deal with the entire training set for each iteration thus they cannot efficiently treat large or dynamically changing data. Online dictionary learning algorithms processes one signal at a time or in small sets and have a faster convergence w.r.t. classical DL methods. ODL [35] is one of the first online dictionary learning algorithm and it set a standard on the top of which many other online strategies have developed.

3.4.2.1 ODL

ODL is an interesting alternative for inferring a dictionary from large training sets or ones which change over time [35], like K-SVD this algorithm also updates the entire dictionary sequentially, but draws one element of training data at a time for the dictionary update. ODL assumes that the training set $\mathbf{Y}$ is composed of i.i.d. samples of a distribution $\mathbf{p}(\mathbf{x})$. In many practical applications, $\mathbf{Y}$ can be obtained by a collection of measurements that has been randomly permuted and will be drawn consecutively for each iteration (as it happens for our work, see chapter 4).
The input parameters of the algorithm are: the initial dictionary $\mathbf{D}_0 \in \mathbb{R}^{M\times K}$, the regularization parameter λ, the dimension of the learned dictionary K, and the number of iterations T (which will also correspond to the number of training set elements that will be used for learning $\mathbf{D}$). The first step at the iteration t, with with $t = 1...T$, is to draw an example of the training set $\mathbf{y}_i$ from $\mathbf{Y}$, then the sparse decomposition step is done using the dictionary obtained at the previous operation $\mathbf{D}_{(t-1)}$ via a Cholesky-based implementation of the LARS-Lasso algorithm [82] which solves a l_1-regularized least-squares problem as indicated in 3.30. In the dictionary update we consider all the training set elements analyzed so far: $\mathbf{y}_i$ with $i = 1...t$.

$$\hat{\mathbf{x}}_{(t)} = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}||\mathbf{y}_t - \mathbf{D}_{(\mathbf{t-1})}\mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1, \tag{3.30}$$

The next step is the dictionary update, this step requires the input dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k] \in \mathbb{R}^{K\times K}$ and two matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k] \in \mathbb{R}^{M\times K} = \sum_{i=1}^t \mathbf{x}_i\mathbf{x}_i^T$ and $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k] \in \mathbb{R}^{M\times K} = \sum_{i=1}^t \mathbf{y}_i\mathbf{x}_i^T$. This algorithm updates each column of $\mathbf{D}$ sequentially using block coordinate descent with a "warm restart" (which consists in the dictionary calculated at the previous step $\mathbf{D}_{t-1}$). The following equations are used for updating the jth column of $\mathbf{D}$ while keeping the other ones fixed:

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{A}_{jj}}(\mathbf{b}_j - \mathbf{D}\mathbf{a}_j) + \mathbf{d}_j \tag{3.31}$$

$$\mathbf{d}_j \leftarrow \frac{1}{max\left(||\mathbf{u}_j||_2\right)}\mathbf{u}_j \tag{3.32}$$

This approach gives the solution to the dictionary update problem:

$$\mathbf{D}_t \triangleq \underset{\mathbf{D}\in C}{\text{argmin}}\ \frac{1}{t}\left(\sum_{i=1}^{t}||\mathbf{y}_i - \mathbf{D}\mathbf{x}_i||_2^2 + \lambda||x_i||_1\right) = \tag{3.33}$$

$$\underset{\mathbf{D}\in C}{\text{argmin}}\ \frac{1}{t}\left(\frac{1}{2}\text{Tr}\left(\mathbf{D}^T\mathbf{D}\mathbf{A}_t\right) - \text{Tr}\left(\mathbf{D}^{\mathbf{T}}\mathbf{B}_t\right)\right) \tag{3.34}$$

Where **C** is a convex set, i.e. we have to impose that the column of the dictionary matrix have an l_2-norm less than or equal to one. $C \triangleq \left\{\mathbf{D} \in \mathbb{R}^{M\times K} \text{ subject to } \forall\ j = 1, ..., k\ \mathbf{d}_j^T\mathbf{d}_t \leq 1\right\}$. The ODL algorithm is faster that the batch K-SVD and, since it uses $\mathbf{D}_{(\mathbf{t-1})}$ as warm restart for computing $\mathbf{D}_\mathbf{t}$, few iterations could be enough for the correct reconstruction. We refer the reader to [35] for further insights on ODL.

A few improvements to ODL have already been proposed. For example, [101] considered a faster Online Sparse Dictionary Learning (OSDL) to efficiently handle bigger training set dimensions using a double-sparsity model.

3.4.2.2 CBWLSU

A recent study [73] notes that even though online processing reduces computational complexity compared to batch methods, ODL performance can be further improved if only useful information from previous data is used for updating the atoms. In this study, a new online DL called Correlation-Based Weighted Least Square Update (CBWLSU) was proposed. CBWLSU is an online method that introduces an interesting alternative for the dictionary update step. Like ODL, CBWLSU evaluates one new training data at a time, $\mathbf{y}_t$. However, to update the dictionary, it searches among all previous training data and uses only the ones which share the same atoms with $\mathbf{y}_t$. Let $\mathbf{Y}_{\mathcal{Q}_t} = \begin{bmatrix}\mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{|\mathcal{Q}_t|}}\end{bmatrix}$, where $l_1, \cdots, l_{|\mathcal{Q}_t|} \in \mathcal{Q}_t$. be the set of previous training elements at iteration t. Define $\mathcal{N}_t = \left\{l : 1 < l < t, \langle\mathbf{x}_l^T, \mathbf{x}_t\rangle \neq 0\right\} \subset \mathcal{Q}_t$ as the set of indices of all previous training elements that are correlated with the new element such that $|\mathcal{N}_t| = N_{p_t}$. The new index set is $\mathcal{N}_t = \mathcal{N}_t \cup \{t\}$ so that the training set becomes $\mathbf{Y}_{\mathcal{N}_t} = \begin{bmatrix}\mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{|\mathcal{N}_t|}}\end{bmatrix}$, where $l_1, \cdots, l_{|\mathcal{N}_t|} \in \mathcal{N}_t$. CBWLSU then employs a weighting matrix $\mathbf{W}(\mathbf{y}_t)$ to evaluate the influence of the selected previous elements for the dictionary update step and solves the optimization problem therein via weighted least squares (WLS). The sparse coding in CBWLSU is achieved via batch OMP.

3.4.2.3 DOMINODL

We now introduce our DOMINODL approach for online DL which not only leads to a dictionary (**D**) that is tuned to sparsely represent the training set (**Y**) but is also faster than other online algorithms. The key idea of DOMINODL is as follows: When sequentially analyzing the training set, it is pertinent to leverage the memory of previous data in the dictionary update step. However, algorithms such as CBWLSU consider *all* previous elements. Using all previous training set samples is computationally expensive and may also slow down convergence. The samples which have already contributed in the dictionary update do not need to be considered again. Moreover, in some real-time applications (such as highly correlated range profiles of GPR), their contribution may not be relevant anymore for updating the dictionary.

In DOMINODL, we save computations by considering only a small batch of previous elements that are correlated with the new elements. The two sets are defined correlated if, in their sparse decomposition, they have at least one common non-zero element. The time gained from considering fewer previous training elements is used to consider a mini-batch of new training data (instead of a single element as in ODL and CBWLSU).

The sparse coding step of DOMINODL employs batch OMP, selecting the maximal residual error δ in (3.23) using a data-driven entropy-based strategy as described later in this section. At the end of each iteration, DOMINODL also drops-off those previous training set elements that have not been picked up after a certain number of iterations, N_u. The mini-batch drawing combined with dropping off training elements and entropy-based criterion to control sparsity results in an extremely fast online DL algorithm that is beneficial for real-time radar operations.

We initialize the dictionary $\mathbf{D}$ using a collection of K training set samples that can be randomly chosen from $\mathbf{Y}$ (alternatively one can also use random vectors with a given distribution) and then perform a sparse decomposition of $\mathbf{Y}$ with the dictionary $\mathbf{D}$; the algorithm then scans the entire training set sequentially. Define the mini-batch of $N_b + 1$ new training elements as $\mathbf{Y}_{\mathcal{B}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{N_b+1}} \end{bmatrix}$ with $l_1, \cdots, l_{N_b+1} \in \mathcal{B}_t$ such that the index set $\mathcal{B}_t = \{l : t \leq l < t + N_b\}$. When $t > L - N_b$, we simply take the remaining new elements to constitute this mini-batch[1]. We store the set of dictionary atoms participating in the SR of the signals in $\mathbf{Y}_{\mathcal{B}_t}$ as $\mathbf{D}_{\mathcal{B}_t}$. Let the coefficient vectors associated with the SR of $\mathbf{Y}_{\mathcal{B}_t}$ are indicated with $\mathbf{X}_{\mathcal{B}_t}$ and

$$\mathbb{I}_{\mathbf{X}_{\mathcal{B}}} = \sum_{p \in \mathcal{B}_t} \mathbf{x}_p \tag{3.35}$$

being an indicator vector whose non-zero elements indicate the atoms of $\mathbf{D}$ being used by $\mathbf{Y}_{\mathcal{B}_t}$.

Define $\mathbf{Y}_{\mathcal{Q}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_t} \end{bmatrix}$ with $l_1, \cdots, l_t \in \mathcal{Q}_t$ the collection of previous training elements with the index set $\mathcal{Q}_t = \{l : 1 \leq l < t - 1\}$. Consider $\mathbf{Y}_{\mathcal{M}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{N_r}} \end{bmatrix}$ with $l_1, \cdots, l_{N_r} \in \mathcal{M}_t \subset \mathcal{Q}_t$ as a randomly selected mini-batch of N_r previous elements. The coefficient vectors associated with the SR of $\mathbf{Y}_{\mathcal{M}_t}$ are indicated with $\mathbf{X}_{\mathcal{M}_t}$ and

$$\mathbb{I}_{\mathbf{X}_{\mathcal{M}}} = \sum_{l \in \mathcal{M}_t} \mathbf{x}_l \tag{3.36}$$

is an indicator vector whose non-zero elements indicate the atoms of $\mathbf{D}$ being used by $\mathbf{Y}_{\mathcal{M}_t}$.

Define $\mathbf{Y}_{\mathcal{A}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{|\mathcal{A}_t|}} \end{bmatrix}$ with $l_1, \cdots, l_{|\mathcal{A}_t|} \in \mathcal{A}_t$ where

$$\mathcal{A}_t = \left\{ l : l \in \mathcal{M}_t, \left\langle {\mathbb{I}_{\mathbf{X}_{\mathcal{B}_t}}}^T, \mathbb{I}_{\mathbf{X}_{\mathcal{M}_t}} \right\rangle \neq 0 \right\} \subset \mathcal{M}_t, \tag{3.37}$$

is a subset of previous training elements that are correlated with the mini-batch of new elements. In order to avoid multiple occurrences of the same element in consecutive mini-batches, DOMINODL ensures that $\mathcal{M}_t \cap \mathcal{M}_{t-1} = \emptyset$ providing that a sufficient number of previous training set elements

[1] In numerical experiments, we observed that the condition $t > L - N_b$ rarely occurs because DOMINODL updates the dictionary and converges in very few iterations. The algorithm also ensures that the number of previous samples $\geq 2N_r$ before the dictionary update. If this condition is not fulfilled, then it considers all previous training samples.

is available. Let $\mathbf{D}_{\mathcal{A}_t}$ be the set of dictionary atoms used for SR of $\mathbf{Y}_{\mathcal{A}_t}$. Our new training set is $\mathbf{Y}_{\mathcal{C}_t} = \mathbf{Y}_{\mathcal{A}_t} \cup \mathbf{Y}_{\mathcal{B}_t}$. Both mini-batches of new and previous elements are selected such that the entire training set size $(N_b + N_r)$ is still smaller than that of CBWLSU where it is $N_{p_t} + 1$ (see above on CBWLSU explanation).

The dictionary update subproblem then reduces to considering only the sets $\mathbf{Y}_{\mathcal{C}_t}$, $\mathbf{D}_{\mathcal{C}_t}$ and $\mathbf{X}_{\mathcal{C}_t}$:

$$\hat{\mathbf{D}}_{\mathcal{C}_t} = \underset{\mathbf{D}_{\mathcal{C}_t} \in \mathcal{D}}{\text{argmin}} \ ||\mathbf{Y}_{\mathcal{C}_t} - \mathbf{D}_{\mathcal{C}_t}\mathbf{X}_{\mathcal{C}_t}||_F^2. \tag{3.38}$$

Assume that the sparse coding for each example is known and define the errors as

$$\mathbf{E}_{\mathcal{C}_t} = \mathbf{Y}_{\mathcal{C}_t} - \mathbf{D}_{\mathcal{C}_t}\mathbf{X}_{\mathcal{C}_t} = [\mathbf{e}_1, \cdots, \mathbf{e}_{N_r}]. \tag{3.39}$$

We can update $\mathbf{D}_{\mathcal{C}_t}$, such that the above error is minimized, with the assumption of fixed $\mathbf{X}_{\mathcal{C}_t}$. A similar problem is considered in MOD where error minimization is achieved through least squares. Here, we employ weighted least squares inspired by the fact that it has shown improvement in convergence over standard least squares [73]. We compute the weighting matrix $\mathbf{W}_{\mathcal{C}_t}$ using the sparse representation error $\mathbf{E}_{\mathcal{C}_t}$

$$\mathbf{W}_{\mathcal{C}_t} = \text{diag}\left(\frac{1}{||\mathbf{e}_1||_2^2}, \ldots, \frac{1}{||\mathbf{e}_{N_r}||_2^2}\right), \tag{3.40}$$

and then solve the following optimization problem

$$\hat{\mathbf{D}}_{\mathcal{C}_t} = \underset{\mathbf{D}_{\mathcal{C}_t} \in \mathcal{D}}{\text{argmin}} \ ||(\mathbf{Y}_{\mathcal{C}_t} - \mathbf{D}_{\mathcal{C}_t}\mathbf{X}_{\mathcal{C}_t})\mathbf{W}_{\mathcal{C}_t}^{\frac{1}{2}}||_F^2. \tag{3.41}$$

This leads to the weighted least squares solution

$$\hat{\mathbf{D}}_{\mathcal{C}_t} = \mathbf{Y}_{\mathcal{C}_t}\mathbf{W}_{\mathcal{C}_t}\mathbf{Y}_{\mathcal{C}_t}^T(\mathbf{Y}_{\mathcal{C}_t}\mathbf{W}_{\mathcal{C}_t}\mathbf{Y^T}_{\mathcal{C}_t})^{-1}. \tag{3.42}$$

The dictionary $\mathbf{D}$ is then updated with the atoms $\hat{\mathbf{D}}_{\mathcal{C}_t}$ and its columns are normalized by their ℓ_2-norms.

The $\mathbf{D}$ is then used for updating the sparse coding of $\mathbf{Y}_{\mathcal{C}_t}$ using batch OMP.Algorithm 4 summarizes all major steps of DOMINODL.

Table 3.1 summarizes the important differences between DOMINODL and other related algorithms. Like MOD and CBWLSU, DOMINODL uses weighted least squares solution in the dictionary update. The proof of convergence for the alternating minimization method in MOD was provided in [102] where it is shown that alternating minimization converges linearly as long as the following assumptions hold true: sparse coefficients have bounded values, sparsity level is on the order of $\mathcal{O}(M^{1/6})$ and the dictionary satisfies the RIP property. In [73], these assumptions have been applied for CBWLSU convergence. Compared to CBWLSU, the improvements in DOMINODL include mini-batch based data selection and data reduction via drop-off strategy but the update algorithms remain the same. Numerical experiments in Chapter 4 suggest that DOMINODL usually converges in far fewer iterations than CBWLSU.

Algorithm 4: Drop-Off MINi-Batch Online Dictionary Learning (DOMINODL)

```
Input: Training set ($\mathbf{Y}$), number of trained atoms ($K$), mini-batch dimension for new training data ($N_b$), mini-batch dimension for previous training data ($N_r$), drop-off value ($N_u$), convergence threshold ($\chi \in \mathbb{R}$) and Residual error threshold ($\delta$) for SR
Output: Learned dictionary ($\mathbf{D}$), sparse decomposition of the training set ($\mathbf{X}$)
Generate the initial dictionary $\mathbf{D}$ of dimension $K$ using training samples
Normalize the columns of $\mathbf{Y}$ and $\mathbf{D}$ by their $\ell_2$-norms
Sparsely decompose $\mathbf{Y}$ with the initial dictionary using batch OMP
Loop
    Gather a mini-batch of new training set elements $\mathbf{Y}_{\mathcal{B}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{N_b+1}} \end{bmatrix}$ with $l_1, \cdots, l_{N_b+1} \in \mathcal{B}_t$ such that the index set $\mathcal{B}_t = \{l : t \leq l < t + N_b\}$
    SR of $\mathbf{Y}_{\mathcal{B}_t}$ with the dictionary $\mathbf{D}$ using entropy-thresholded batch OMP
    Store the set of atoms $\mathbf{D}_{\mathcal{B}_t}$ participating in the SR of $\mathbf{Y}_{\mathcal{B}_t}$
    $\mathbf{X}_{\mathcal{B}_t} \leftarrow$ coefficient vectors associated with the SR of $\mathbf{Y}_{\mathcal{B}_t}$ and indicator vector $\mathbb{I}_{\mathbf{X}_{\mathcal{B}}} \leftarrow \sum_{p \in \mathcal{B}_t} \mathbf{x}_p$
    $\mathbf{Y}_{\mathcal{Q}_t} \leftarrow \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_t} \end{bmatrix}$ with $l_1, \cdots, l_t \in \mathcal{Q}_t$ as the collection of previous training elements with the index set $\mathcal{Q}_t = \{l : 1 \leq l < t - 1\}$
    Randomly select a mini-batch of previous training set elements $\mathbf{Y}_{\mathcal{M}_t} = \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{N_r}} \end{bmatrix}$ with $l_1, \cdots, l_{N_r} \in \mathcal{M}_t \subset \mathcal{Q}_t$
    $\mathbf{X}_{\mathcal{M}_t} \leftarrow$ the coefficient vectors associated with the SR of $\mathbf{Y}_{\mathcal{M}_t}$ and indicator vector $\mathbb{I}_{\mathbf{X}_{\mathcal{M}}} \leftarrow \sum_{l \in \mathcal{M}_t} \mathbf{x}_l$
    $\mathbf{Y}_{\mathcal{A}_t} \leftarrow \begin{bmatrix} \mathbf{y}_{l_1} & \cdots & \mathbf{y}_{l_{|\mathcal{A}_t|}} \end{bmatrix}$ with $l_1, \cdots, l_{|\mathcal{A}_t|} \in \mathcal{A}_t$, where $\mathcal{A}_t = \left\{ l : l \in \mathcal{M}_t, \left\langle {\mathbb{I}_{\mathbf{X}_{\mathcal{B}_t}}}^T, \mathbb{I}_{\mathbf{X}_{\mathcal{M}_t}} \right\rangle \neq 0 \right\} \subset \mathcal{M}_t$,
    $\mathbf{Y}_{\mathcal{C}_t} \leftarrow \mathbf{Y}_{\mathcal{A}_t} \cup \mathbf{Y}_{\mathcal{B}_t}$ and store $\mathbf{D}_{\mathcal{C}_t}$ the atoms of $\mathbf{D}$ shared by $\mathcal{B}_t$ and $\mathcal{M}_t$
    $\mathbf{E}_{\mathcal{C}_t} \leftarrow \mathbf{Y}_{\mathcal{C}_t} - \mathbf{D}_{\mathcal{C}_t} \mathbf{X}_{\mathcal{C}_t} = [\mathbf{e}_1, \cdots, \mathbf{e}_{N_r}]$.
    $\mathbf{W}_{\mathcal{A}_t} \leftarrow \mathrm{diag} \left( \frac{1}{||\mathbf{e}_1||_2^2}, \ldots, \frac{1}{||\mathbf{e}_{N_r}||_2^2} \right)$
    $\hat{\mathbf{D}}_{\mathcal{C}_t} \leftarrow \hat{\mathbf{D}}_{\mathcal{C}_t} = \mathbf{Y}_{\mathcal{C}_t} \mathbf{W}_{\mathcal{C}_t} \mathbf{Y}_{\mathcal{C}_t}^T (\mathbf{Y}_{\mathcal{C}_t} \mathbf{W}_{\mathcal{C}_t} \mathbf{Y}_{\mathcal{C}_t}^T)^{-1}$ and normalize its columns
    Replace the updated atoms $\mathbf{D}_{\mathcal{C}_t}$ into $\mathbf{D}$ and normalize its columns
    Perform SR of selected signals used in the previous step using entropy-thresholded batch OMP
    Eliminate previous training set elements which have not been used for the last $N_u$ iterations
    if $|| \left( \mathbf{Y}_{\mathcal{C}_t} - \mathbf{D}_t \mathbf{X}_{\mathcal{C}_t} \right) (\mathbf{W}_i)^{0.5} ||_F^2 < \chi$ then break
EndLoop
```

Table 3.1 – Comparison of DL steps

DL step	K-SVD	LRSDL	ODL	CBWLSU	DOMINODL
Training method	Batch	Batch	Online	Online	Online
Sparse coding method	OMP	FISTA	LARS	Batch OMP	batch OMP
Dictionary update	Entire **D**	Entire **D**	Entire **D**	Entire **D**	Partial **D** adaptively
Samples per iteration	Entire **Y**	Entire **Y**	$\mathbf{Y}_t$	$\mathbf{Y}_{\mathcal{N}_t}$	$\mathbf{Y}_{\mathcal{C}_t}$
Optimization method	SVD	ADMM	Gradient descent	Weighted least squares	Weighted least squares
Dictionary pruning	Yes	No	Yes	No	No
Training-set drop-off	No	No	No	No	Yes

Although we developed and tested DOMINODL on a highly correlated GPR dataset (see chapter 4), this technique may be employed in other applications where real-time learning is necessary and the signals are correlated. Our tests demonstrate that DOMINODL converges faster than other online DL approaches (see chapter 4) because of the combined strategy of drawing more new elements for each iteration, considering less previous elements in search for correlation and dropping off the unused previous elements.

Computational complexity of DOMINODL has a very low order compared to other online approaches. Let us indicate the order of complexity as OC_{DL}, with the subscript DL indicating the employed DL algorithm. As mentioned earlier, there are N atoms in the dictionary. Assume that every signal is represented by a linear combination of K_s atoms, $K_s \ll N$. Empirically, among all possible combinations of K_s atoms from N, the probability to have a common atom in the sparse representation is K_s/N. Given L training elements, the number of training data which have a specific atom in their representation is proportional to LK_s/N. Suppose our mini-batch has elements that reduce the number of training data by a factor $\beta < 1$ (depending on the value of N_b and N_r. Further, assume that the dropping off step reduces the training set elements by a factor $\rho < 1$. The number of used training data L_t in the t^{th} iteration is proportional to $\beta\rho t K_s/N$. Then, the worst estimate of DOMINODL's computational complexity is due to the sparse coding batch OMP which is of order $OC_{DOMINODL} = \mathcal{O}(L_t N^2) = \mathcal{O}(\beta\rho t K_s N) \approx \mathcal{O}(\beta\rho t N)$. This is much smaller than the complexity of ODL ($OC_{ODL} = \mathcal{O}(N^3)$) or CBWLSU ($OC_{CBWLSU} = \mathcal{O}(tN)$) [35, 73].

Figure 3.1 illustrates the computational complexity of online DL approaches compared to DOMINODL. Here, we generally indicate with $OC_{DL}(t)$ and $OC_{DL}(k)$ the order of complexity in function of the number of iterations and atoms of the dictionary. Figure 3.1(a) shows that, for fixed number of iterations ($t = 60$), the general trend of complexity with respect to the increase in the number of atoms (K) is similar for all algorithms. However, the complexity of ODL is higher than CBWLSU and DOMINODL; the latter being the least complex. When the number of iterations is increased, the complexity of ODL and CBWLSU have a similar increasing trend (see Fig. 3.1(b)). In case of DOMINODL, its complexity is similar to the increasing trend of CBWLSU and determined largely by N_b. When DOMINODL iterations begin accounting for N_r previous elements, its complexity stays constant. The value of β changes for every iteration, while ρ depends on the data itself. In general, after a few dozen of iterations, DOMINODL's complexity always stays lower than CBWLSU.

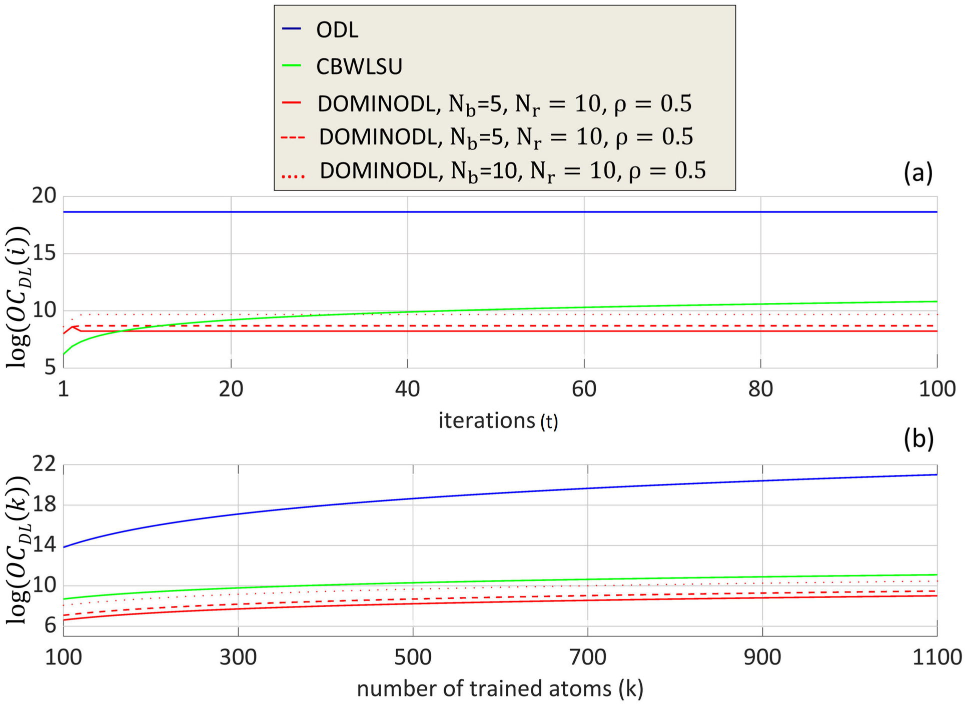


Figure 3.1 – Computational complexity of online DL strategies for increasing number of (a) iterations and (b) trained atoms.

The proposed DL methods are heuristic, they do yield over-complete dictionaries but with no theoretical guarantee of obtaining coherent dictionaries. Several state-of-the-art results that outline DL algorithms with concrete performance guarantees require stronger assumptions on the observed data.

An efficient sparse representation (SR) which accurately represents the scattering behaviors related to soil type and targets serves a twofold purpose. First, it allows application of CS in reducing sensing data and associated scan time. Second, the literature [69, 103] indicates that SR is effective in extracting the mid- or high-level features in image classification. For example, in the context of landmines classification with Ground Penetrating Radar, our prior work [40, 104] has shown that frameworks based on sparse representation improve the classifier performance. SR has been frequently applied to data from synthetic aperture radar (SAR) [105]), inverse SAR (ISAR) [36]) and interferometric SAR (InSAR) [106] because these are not naturally sparse in the range-time domain. In [107], it has been shown that an efficient SR for SAR image can lead to reducing computational load in signal reconstruction from partial measurements. In the particular case of GPR, [14, 108–110] have proposed CS-based imaging for various GPR waveforms.

Techniques to reduce the aquired radar samples using CS have been previously used to address the bottleneck of long scan times in conventional radar systems. The estimation accuracy of target parameters is greatly affected by radar's dwell time [111], i.e., the time duration a directional radar beam spends hitting a particular target. But, at the same time, this negatively affects the ability of the radar to look at targets in other directions thereby prolonging the total scan time. For example, [112] employs matrix completion to reduce scan time in a weather radar. In [113], Xampling framework in slow-time domain is used to recover target parameters with fewer transmit pulses. Similar techniques have also been shown to be useful in radar imaging applications [114].

Chapter 4

Online dictionary learning for adaptive GPR target recognition

In this section, we present a strategy for buried landmine classification using GPR range profiles (i.e A-Scans). The proposed methodology is based on the combined application of different DL algorithms (for improved sparse representation) and Support Vector Machine (for classification). More particularly, we apply batch and online DL methods for generating dictionaries apt to sparsely represent our GPR measurements. The resulting sparse vectors are then employed as an input to a Support Vector Machine (SVM) classifier (see see Appendix B for an explanation of how SVM works) which discriminates between different types of mines and clutter. The results of the proposed approach will be evaluated on experimental measurements from a GPR test field with buried landmine-simulants. We will utilize all the DL algorithms described in chapter 3, including our novel Drop-Off MINibatch Online Dictionary Learning (DOMINODL).

The selection of the different parameters for the presented DL algorithms is crucial for their success in sparsifying the data. Contrary to previous studies which determine DL parameters (number of iterations, atoms, etc.) based on bulk statistics such as normalized root-mean-square-error (NRMSE) [41], we consider statistical inference for parameter analysis. Our methods are based on Kolmogorov-Smirnoff test distance [115] and Dvoretzky-Kiefer-Wolfowitz (DKW) Inequality [116, 117]

The comparison of K-SVD with three online DL algorithms - ODL, its correlation-based variant [73] and DOMINODL - shows that online methods present distinct advantages in speed (in particular DOMINODL) and low false-alarm rates, successfully improving the detection of mines with very small RCS buried deep into clutter and noise.

Some recent studies [3, 118–120] employ state-of-the-art deep learning approaches such as a convolutional neural network (CNN) to classify GPR-based mines data (see Appendix A) for an explanation of how CNN works). A main limitation of these kinds of approaches is the necessity of large training sets. The comparison with CNN illustrates that it has poorer performance in detection of small mines than our online DL approaches. This may also be caused by the relatively small dimensions of our training set which, even if sufficient for DL, may not meet the expected requirements for CNN [121]. We also show that the classification performance of online DL methods does not deteriorate seriously

when signal samples are reduced. This may be specially convenient for application such as mine detection, where time and computational cost
This chapter is organized as follows. We initially provide an overview of the GPR system, field campaign and collected data sets. We then introduce our techniques for DL parameter selection and finally we present classification results.

4.1 Experimental measurements

In this section, we first provide details of our GPR system and the field measurement campaign. We then describe the procedure to organize the entire dataset for our application.

4.1.1 Ground Penetrating Radar System

The GPR system(see Fig. 4.1) is the commercially available Surface Penetrating Radar unit called *SPRScan* manufactured by ERA Technology. It is an L-band, impulse waveform, ultra-wideband (UWB) radar that is mounted on a movable trolley platform (see figure 4.1). Pulsed GPRs are more effective in terms of offering penetration depth and wide bandwidth with respect to the standard Stepped-Frequency Continuous Wave (SFCW) systems. The former is also more robust to electronic interference and does not suffer from unequal balancing of antenna signals [122].

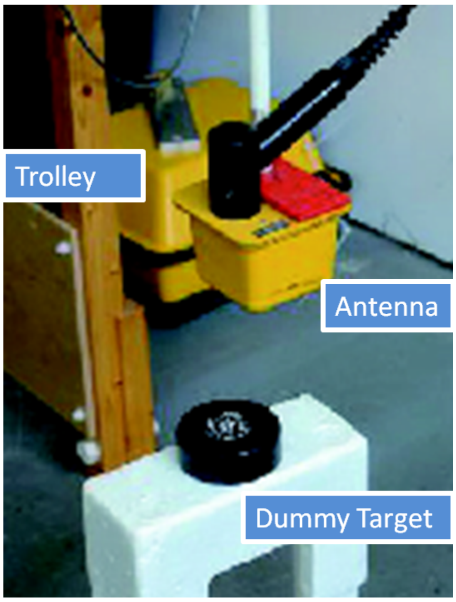


Figure 4.1 – The L-band GPR system is attached to a movable trolley platform. It is mounted along a rail system and scans the target from above.

Table 4.1 lists the most important operational parameters of the system. The radar uses a 8×8 cm dual bow-tie dipole antenna for both transmit (Tx) and receive (Rx) sealed in a metallic shielding filled with an internal absorber. The central frequency of the system (f_c) and its bandwidth (Δf) are 2 GHz. The pulse repetition frequency (PRF) and the sampling of the receiver ADC is 1 MHz. The scanning system has a resolution of 1 cm towards the perpendicular broadside (or X direction) and 4 cm towards the cross-beam (Y direction). In our field campaigns, the SPRScan system moves along

Table 4.1 – Technical characteristics of impulse GPR

Parameter	Value
Operating frequency	2 GHz
Pulse repetition frequency	1 MHz
Pulse length	0.5 ns
Sampling time	25 ps
Spatial sampling along the beam	1 cm
Cross-beam resolution	4 cm
Antenna height	5-9 cm
Antenna configuration	Perpendicular broadside
Samples/A-scan	512

the survey area over a rail system which allows accurate positioning of the sensor head in order to obtain the aforementioned resolution in X and Y (see also Section 2.2).

The transmit pulse of the GPR system is a *monocycle* (see 2.2). The scattering of UWB radar signals from complex targets that are composed of a finite number of scattering centers can be described in terms of the channel impulse response (CIR). Here, the CIR is considered as a linear, time invariant, causal system which is a function of the target shape, size, constituent materials, and scan angle. The CIR $h(t)$ of a GPR target, with M scatterers, is expressed as a series of time-delayed and weighted Gaussian pulses [123]

$$h(t) = \sum_{m=1}^{M} \alpha_m e^{-4\pi[(t-t_m)/\Delta T_m]^2}, \tag{4.1}$$

where each scatterer located at range r_m from the radar is characterized by the reflectivity α_m, duration ΔT_m, relative time shift $t_m = 2r_m/v_s$, where $v_s = c/\sqrt{\epsilon_r}$ is the speed of the electromagnetic wave in the soil, $c = 3 \times 10^8$ m/s is the speed of light, and ϵ_r is the dielectric constant which depends on the soil composition and moisture.

The response of the target to the Gaussian monocycle is the received signal

$$y(t) = s_T(t) * h(t), \tag{4.2}$$

also regarded as the *range profile*.

For each X/Y position, the system receives a radar echo (range profile) from the transmitted pulse. In order to deal with the exponential signal attenuation during the propagation through the soil medium, the dynamic range of the signal is enhanced via stroboscopic sampling [124–126]. This technique comprises integrating N receiver samples (generated by transmitting a sequence of N pulses) at the ADC receiver sampling rate but with a small time offset δ for each of them. To achieve the desired stroboscopic sampling rate T_s, the time offset must be selected accordingly, i.e., $\delta = T_s/N$ [126]. Our GPR system employs stroboscopic sampling to reach a pseudo sampling frequency of $f_s = 1/T_s = 40$ GHz (much above the Nyquist rate) to yield the discrete-time signal $y[n] = y(nT_s)$. The receiver has the ability to acquire a maximum of 195 profiles per second, each one consisting of 512 range samples. Prior to the A/D conversion, the signal is averaged to improve the signal to noise ratio (SNR). A time-varying gain correction can be applied to compensate for the soil attenuation and increase the overall dynamic range of the system. The receiver averages 100 range profiles for each antenna position.

4.1.2 Test Field

We evaluated the proposed approach with the measurement data from a 2013 field campaign at Leibniz Institute for Applied Geophysics (LIAG) in Hannover (Germany) [11]. Fig. 4.2 shows the test field, for detailed ground truth informations.

The soil texture was sandy and highly inhomogeneous (due to the presence of material such as organic matter and stones), thereby leading to a high variability in the electrical parameters. The dielectric constant at three different locations of the testbed was measured with a Time Domain Reflectometer (TDR) to obtain an estimate of its mean value and variability. The average value oscillated between 4.6 and 10.1 with 15% standard deviation and *correlation length* [11] of 20 cm. These big variations in soil dielectric properties pose difficulties in mine detection. During the field tests, the SPRScan system moved on two plastic rails with the scan resolution in the X and Y directions being 1 and 4 cm, respectively. The entire survey lane was divided in 1×1 m sections (see Fig. 4.2), each containing two targets in the center. The targets on the left and right sides of the lane were buried at approximately 10 and 15 cm depths, respectively.

Our testbed contains standard test targets (STT) and simulant landmines (SIM) of different sizes and shapes. An STT is a surrogate target used for testing landmine detection equipment. It is intended to interact with the equipment in an identical manner as a real landmine does. An SIM has the representative characteristics of a specific landmine class although it is not a replica of any specific model [127]. In this paper, we study three STTs (PMA2, PMN and Type-72) and one SIM (ERA). All of these test objects are buried at a depth of 10-15 cm in the test field [2]. For classification purposes, we group PMN and PMA2 together as the largest targets while T72 mines are the smallest (Fig. 4.3).

4.1.3 Dataset Organization

The entire LIAG dataset consists of 27 survey sections (or simply, ”surveys”, see 4.2) of size 1×1 m. Every survey consists of 2500 range profiles. We divided the data into two sets: The *training set* ($\mathbf{Y}$) to be used for both DL and classification and the *test set* ($\mathbf{Y}_{\text{TEST}}$): to evaluate the performance of the classification.

The training set $\mathbf{Y} \in \mathbb{R}^{M \times L}$ is a matrix whose L columns $\{\mathbf{y}_i\}_{i=1}^{L}$ consist of sampled range profiles $\mathbf{y}_i = \left[y[0], \cdots, y[M-1]\right]^T$ of M range samples each. The profiles are selected from different surveys and contain almost exclusively either a particular class of landmine or clutter. In total, we have 463, 168, 167 and 128 range profiles for clutter, PMA2/PMN, ERA and Type-72, respectively, see

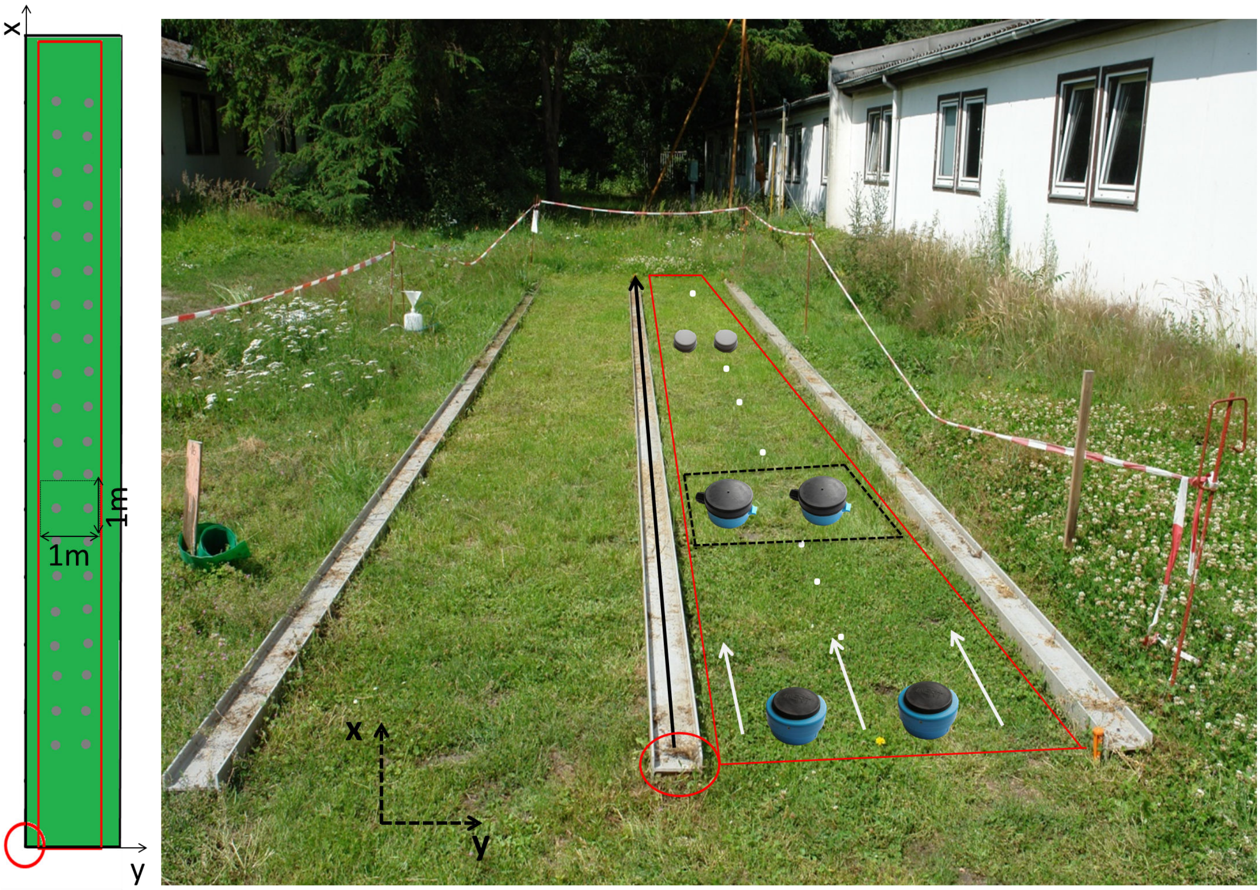


Figure 4.2 – The LIAG test field in Hannover (right) along with its layout (left). The scan directions X and Y of the radar are indicated on the photograph and layout. The radar coverage region is indicated by solid red lines with a red circle showing the origin of the scan. The white arrows in the photograph indicate specific lanes scanned in the X direction that are separated in the Y direction by 4 cm. In the layout, each gray dot represents the location of a buried test target. An individual survey area unit of 1 m × 1 m that contains 2 targets is also indicated on the layout (solid black lines) and the photograph (dotted black lines). The solid black arrow over the middle rail in the photograph is where the SPRScan was mounted.

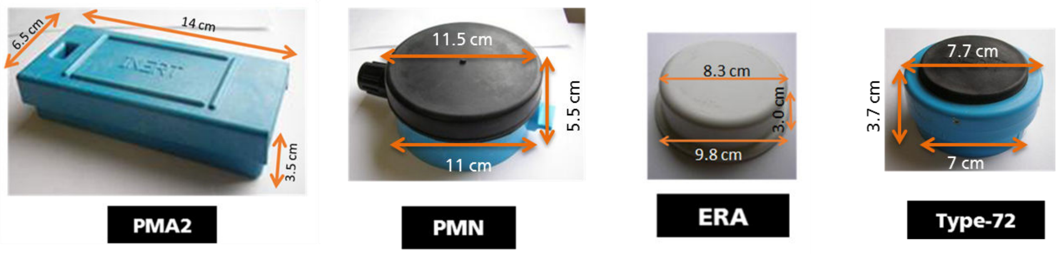


Figure 4.3 – Details of the simulant landmines and the standard test target buried in the test field.

table 4.2. An accurate separation of these classes was very challenging because of the contributions from the non-homogeneous soil clutter that often masked the target responses completely. A poor

Table 4.2 – Training set classes

Target class	Number of elements
Clutter	463
PMN	168
ERA	167
T72	128

selection would lead the DL to learn a dictionary that is appropriate for sparsely representing clutter, instead of landmines. The test set $\mathbf{Y}_{\text{TEST}} \in \mathbb{R}^{M \times J}$ is a matrix with $J = 15000$ columns $\{\mathbf{y}_{\text{TEST}_i}\}_{i=1}^{J}$ that correspond to sampled range profiles from 6 surveys, two for each target class. The test and training sets contain data from separate surveys to enable fair assessment of the classification performance. We denote by the matrices $\mathbf{X} \in \mathbb{R}^{N \times L}$ and $\mathbf{X}_{\text{TEST}} \in \mathbb{R}^{N \times J}$ as the sparse representations of $\mathbf{Y}$ and $\mathbf{Y}_{\text{TEST}}$, respectively and K by the number of atoms of the learned dictionary $\mathbf{D} \in \mathbb{R}^{M \times K}$.

4.1.4 A note on signal pre-processing

We tested different pre-processing techniques on the range profiles coming from our dataset: background subtraction, Fourier transformation, DC component removal, Hilbert transform, etc. We verified the DL and classification performances with the aforementioned processing to note that the best results were obtained with raw-data, time-gated on the area where landmines are usually located adding DC-removal. The time gating helped to remove unwanted contribution from air/ground interface and discard depths where targets are not present, this reduced our signal samples to 512 to 211 corresponding to depths of ca. 5 - 30cm. Techniques such background removal proved to be effective for simulated data (see 2.3.2) but, in real measurements, the landmine response is so close to the background clutter that eliminate the latter means cancelling out the contribution of the mine as well.

4.1.5 Selecting the maximal residual error for SR

Selecting a value for the residual error threshold δ (see 3.2) for SR is usually not straightforward. This value can be related to the amount of noise in the observed data but this information is not known. The samples of our training set can be seen as realizations of a statistical process with an unknown distribution and therefore one can associate to these realizations the concept of *statistical entropy*. We compute the normalized entropy of the mean vector of all the training set samples as

$$E(\mu_{\mathbf{Y}}) = -\sum_{i=1}^{M} P(\mu_{\mathbf{y}_t}) \log P(\mu_{\mathbf{y}_t}), \tag{4.3}$$

where $\mu_{\mathbf{Y}}$ is the mean vector of all training samples, M is the number of features for each training sample and $P(\cdot)$ is the probability mass function. In our case, $P(\cdot)$ is obtained as the normalized histogram of $\mu_{\mathbf{Y}}$. The $E(\mu_{\mathbf{Y}})$ is an indicator of the *randomness* of the data due to noise.

We use the value of $E(\mu_{\mathbf{Y}})$ for the maximal residual error δ for the SR step inside the proposed DL strategies and when performing SR of the test set.

4.2 Parametric evaluation of DL algorithms

The classification performance of our SR/DL based approach is sensitive to the input parameters of DL algorithms thereby making it difficult to directly apply DL with arbitrary parameter values. Previous works set these parameters through hit-and-trial or resorting to metrics that are unable to discriminate the influence of different parameters [41].

In this section, we propose a method to investigate the effect of the various input parameters on the learning performance and then pre-set the parameters to *optimal* values that yield the dictionary $\mathbf{D}$ (for each DL method) optimized to sparsely represent our GPR data, therefore improving the quality of the features for classification (i.e. the sparse coefficients).

Table 4.3 lists these input parameters. K-SVD and ODL take number of iterations N_t, number of trained atoms K as input parameters while CBWLSU uses only K. DOMINODL parameters are the dimension of the mini-batch of new (N_b) and previous (N_r) training elements for each iteration and the drop-off value N_u which indicates after how many iteration the algorithm should discard each unused training set element. We applied K-SVD, ODL, CBWLSU and DOMINODL separately on the training set for different combinations of parameter values.

Table 4.3 – DL parameters

DL algorithm	Input parameters
K-SVD	N_t, K
ODL	N_t, K
CBWLSU	K
DOMINODL	K, N_b, N_r, N_u

In order to compare the dictionaries obtained from various DL algorithms, we use a *similarity measure* that quantifies the closeness of the original training set $\mathbf{Y}$ with the reconstructed set $\hat{\mathbf{Y}}$ obtained using the sparse coefficients of the learned dictionary $\mathbf{D}$. From these similarity values, empirical probability density functions (EPDFs) for any combination of parameter values are obtained; we evaluate these EPDFs using statistical metrics described in Section 4.2.2. These metrics efficiently characterize the similarity between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ and lead us to an optimal selection of various DL input parameters for our experimental GPR dataset.

4.2.1 Similarity Measure

Consider the cross-correlation between a training set vector $\mathbf{y}_i$ and its reconstruction using a learned dictionary $\hat{\mathbf{y}}_i$: The cross-correlation between $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ can be defined as:

$$\mathbf{r}_{\mathbf{y}_i,\hat{\mathbf{y}}_i}[l] = \sum_{n=-\infty}^{+\infty} \mathbf{y}_i[n]\hat{\mathbf{y}}_i[n+l] \tag{4.4}$$

whereas its normalized version (normalized cross-correlation) is given by:

$$\overline{\mathbf{r}_{\mathbf{y}_i,\hat{\mathbf{y}}_i}}[m] = \frac{\mathbf{r}_{\mathbf{y}_i,\hat{\mathbf{y}}_i}[m]}{\sqrt{\mathbf{r}_{\mathbf{y}_i,\mathbf{y}_i}[0]\mathbf{r}_{\hat{\mathbf{y}}_i,\hat{\mathbf{y}}_i}[0]}}. \tag{4.5}$$

For the vector $\mathbf{y}_i$, we finally define the similarity measure s_i as

$$s_i = \max_m |\overline{\mathbf{r}_{\mathbf{y}_i,\hat{\mathbf{y}}_i}(m)}|, \tag{4.6}$$

where a value of s_i closer to unity demonstrates greater similarity of the reconstructed data with the original training set element (i).

We compute $\{s_i\}_{i=1}^{L}$ for all vectors $\{\mathbf{y}_i\}_{i=1}^{L}$, and then obtain the normalized histogram or empirical probability density function (EPDF) of all similarity measures. In statistics, an EPDF is defined as the distribution function associated with the empirical measure of a set of data (in our case a set of similarity measures). We indicate the EPDF of a particular DL algorithm as $p_{s_{DL}}$. Here, the subscript DL represents the algorithm used for learning $\mathbf{D}$ e.g. "K-SVD", "LRSDL", "ODL", "CBWLSU" and "DOMINODL", respectively.

Various parameter combinations for a specific DL method result in a collection of EPDFs. For a given DL method, our goal is to compare the EPDFs by varying these parameters, and arrive at the thresholds of parameter values after which the changes in $p_{s_{DL}}$ are only incremental. For instance, Fig. 4.4 shows the EPDFs of $\{s_i\}_{i=1}^{L}$ obtained from the GPR mines data where optimal parameters for different DL methods were determined using the statistical methods described in the following subsection. In the figure we also indicate which parameters were affecting each particular distributions (in parenthesis). We note that the online DL approaches ($p_{s_{ODL}}$, $p_{s_{CBWLSU}}$ and $p_{s_{DOMINODL}}$) yield distributions that are more skewed towards unity than K-SVD ($p_{s_{K-SVD}}$). The distribution associated to LRSDL is concentrated on a single peak with high values of similarity, nevertheless, it does not exibith the extremely close to unity values of Online-DL distributions.

4.2.2 Statistical Metrics

We are looking for parameter values for which $p_{s_{DL}}$ is skewed towards unity and has small variance. The individual comparisons of mean (μ) and standard deviation (σ) of EPDFs, as used in previous GPR DL studies [41], are not sufficient to quantify the observed dispersion in the EPDFs obtained by varying any of the parameter values. Some DL studies [41, 106, 128] rely on bulk statistics such as the Normalized Root Mean Square Error (NRMSE) but these quantities are less sensitive to changes in parameter values and, therefore, not so helpful in fine-tuning the algorithms. For this evaluation, we will use three different metrics: the coefficient of variation, the Two-sample Kolmogorov-Smirnov (K-S) distance and the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality.

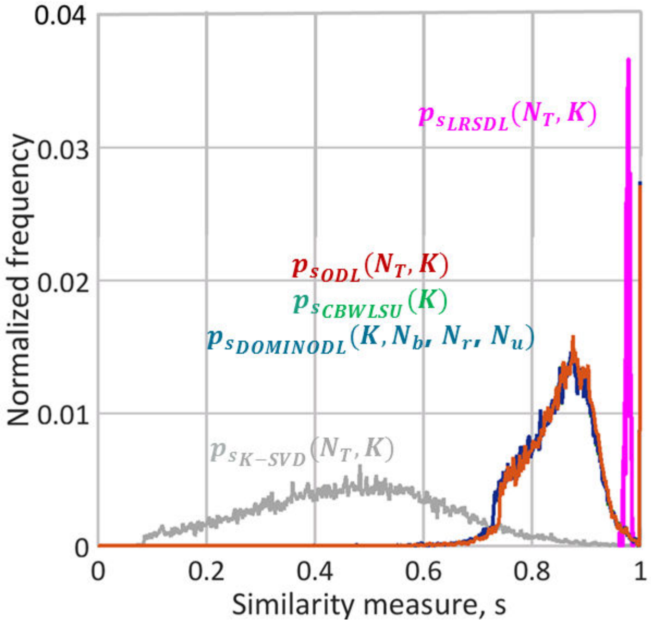


Figure 4.4 – Normalized histograms of similarity measure using the following optimal parameters for the DL algorithm: $N_t = 100$, $K = 640$, $N_b = 30$, $N_r = 10$, and $N_u = 10$. See Section 4.2.3 on the process to select these optimal values.

4.2.2.1 Coefficient of variation

We choose to simultaneously compare both mean (μ) and variance (σ) of a single EPDF by using the *coefficient of variation*:

$$CV = \sigma/\mu \tag{4.7}$$

In our analysis, it represents the extent of variability in relation to the mean of the similarity values.

4.2.2.2 Two-sample Kolmogorov-Smirnov (K-S) distance

In the context of our application, it is more convenient to work with the cumulative distribution functions (CDFs) rather than with EPDFs because the well-developed statistical inference theory allows for convenient comparison of CDFs. Therefore, our second metric to compare similarity measurements obtained by successive changes in parameter values is the *two-sample Kolmogorov-Smirnov (K-S) distance* [115], which is the maximum distance between two given empirical cumulative distribution functions (ECDF). Larger values of this metric indicate that samples are drawn from different underlying distributions. Given two distributions s_1 and s_2 taken at L discrete points, suppose $\hat{F}_{s_1}$ and $\hat{G}_{s_2}$ are their ECDFs of the same length and correspond to their EPDFs $\hat{f}_{s_1}$ and $\hat{g}_{s_2}$, respectively. Denote Ω as the set of L observations used to evaluate both distributions (in our case, L is the number of range profiles in the training set). Then, the discrete two sample K-S distance is:

$$d_{ks}(\hat{F}_{s_1}, \hat{G}_{s_2}) = \sup_{i \in \Omega} |\hat{F}_{s_1}(i) - \hat{G}_{s_2}(i)|, \tag{4.8}$$

where $\sup(\cdot)$ denotes the supremum over all distances.
We first compute a reference ECDF ($\hat{G}_{s_{\text{ref}}}$) for each DL algorithm with fixed parameter values. For

our purposes, this reference ECDF will be obtained by a particular combination of input parameters of the selected DL algorithm. Then, we vary parameter values from this reference and obtain the corresponding ECDF $\hat{F}_{s_{\text{test}}}$ of similarity measure. Finally, we calculate the K-S distance d_{ks} of each $\hat{F}_{s_{\text{test}}}$ with respect to $\hat{G}_{s_{\text{ref}}}$ as

$$d_{ks} = d_{ks}(\hat{F}_{s_{\text{test}}}, \hat{G}_{s_{\text{ref}}}) = \sup_{1 \leq i \leq L} |\hat{F}_{s_{\text{test}}}(i) - \hat{G}_{s_{\text{ref}}}(i)|. \tag{4.9}$$

For our evaluation, d_{ks} states how much the selection of certain input parameters of DL changes the ECDFs of similarity values (i.e. how different is the result of DL) w.r.t. the reference one. In other words, using this metric in combination with CV gives us information regarding the variation and the quality of the obtained dictionary according to the selected input parameters.

4.2.2.3 Dvoretzky-Kiefer-Wolfowitz (DKW) inequality

As a third metric, we exploit the *Dvoretzky-Kiefer-Wolfowitz inequality (DKW)* [116, 117] which precisely characterizes the rate of convergence of an ECDF to a corresponding exact CDF (from which the empirical samples are drawn) for any finite number of samples.
Let $d_{ks}(\hat{G}_s, F_s)$ be the K-S distance between ECDF $\hat{G}_s$ and the continuous CDF F_s for a random variable s and L samples. Since $\hat{G}_s$ changes with the change in the L random samples, $d_{ks}(\hat{G}_s, F_s)$ is also a random variable. We are interested in the conditions that provide desired confidence in verifying if F and G are the same distributions for a given finite L. If the two distributions are indeed identical, then the DKW inequality bounds the probability that d_{ks} is greater than any number ϵ, with $0 < \epsilon < 1$ as follows:

$$\Pr\left\{d_{ks}\left(\hat{G}_s, F\right) > \epsilon\right\} \leq 2e^{-2L\epsilon^2}. \tag{4.10}$$

The corresponding asymptotic result that as $L \to \infty$, $d_{ks} \to 0$ with probability 1 is due to the Glivenko-Cantelli theorem [129, 130].
Consider a binary hypothesis testing framework where we use (4.10) to test the null hypothesis $\mathcal{H}_0 : F = \hat{G}$ for a given ϵ. The DKW inequality bounds the probability of rejecting the null hypothesis when it is true, i.e., the type I statistical error. Assuming that this error is smaller than a certain confidence level α, the following inequality must hold with probability at least $1 - \alpha$ [131]:

$$d_{ks}\left(\hat{G}_s, F\right) \leq \sqrt{-\frac{1}{2L}\ln\left(\frac{\alpha}{2}\right)}. \tag{4.11}$$

Our goal is to use the DKW inequality to compare two ECDFs $\hat{F}_{s_{\text{test}}}$ and $\hat{G}_{s_{\text{ref}}}$ as in (4.9), to verify if they are drawn from the same underlying CDF. By the triangle inequality, the K-S distance

$$d_{ks}(\hat{F}_{s_{\text{test}}}, \hat{G}_{s_{\text{ref}}}) = d_{ks}(\hat{F}_{s_{\text{test}}}, F) + d_{ks}(\hat{G}_{s_{\text{ref}}}, F), \tag{4.12}$$

where G an F are the underlying CDFs corresponding to $\hat{G}$ and $\hat{F}$. We now bound the right side using DKW

$$\begin{aligned} d_{ks}(\hat{F}_{s_{\text{test}}}, \hat{G}_{s_{\text{ref}}}) &\leq \sqrt{-\frac{1}{2L}\ln\left(\frac{\alpha}{2}\right)} + \sqrt{-\frac{1}{2L}\ln\left(\frac{\alpha}{2}\right)} \\ &= \sqrt{-\frac{2}{L}\ln\left(\frac{\alpha}{2}\right)}, \end{aligned} \tag{4.13}$$

which is the maximum distance for which $\hat{F}_{s_{\text{test}}}$ and $\hat{G}_{s_{\text{ref}}}$ are identical with probability $1 - \alpha$. The *DKW metric* is the difference

$$d_{DKW} = \sqrt{-\frac{2}{L}\ln\left(\frac{\alpha}{2}\right)} - d_L(\hat{F}_{s_{\text{test}}}, \hat{G}_{s_{\text{ref}}}). \tag{4.14}$$

Larger values of this metric imply greater similarity betweem the two ECDFs; a negative value implies that the null hypothesis is not true.

For our purposes, d_{DKW} is an alternative way to state the variation of ECDFs respect to d_{KS}, however, d_{DKW} also tell us if two distribution are coming from the same underlying CDFs or not (wether the metric is positive or negative, i.e. fulfilling or not the null hypotesis) thus giving a stronger indication on how the input parameters affect the final result of DL.

4.2.3 Parametric Evaluation results

In this subsection we evaluate the performance of the presented DL algorithms using the metrics explained in 4.2.2 for the reconstruction of the training set $\mathbf{Y}$. There are various soil types and scenarios for a landmine contaminated site. The LIAG test data provides an accurate representation of a practical scenario. Our metrics are general and derived from widely accepted statistical studies. Thus, their relevance to similar scenarios is very likely.

We refer to the Table 4.3 to indicate what are the parameter which affect the presented DL approaches. In particular, the number of iterations N_t is not relevant to CBWLSU and DOMINODL while the latter requires additional parameters for the mini-batch dimensions and the iterations required to drop-off unused training set elements.

Due to the notable amount of information provided by this evaluation, before explaining in detail its results, we give a brief outlook of the final outcomes in table 4.4.

Table 4.4 – Outlook on the influence of the different input parameters for the proposed DL approaches. I = important, S.I. = slightly important, N.U. = not used

	N_t	K	N_b	N_r	N_u
K-SVD	N.I.	I.	N.U.	N.U.	N.U.
ODL	S.I.	I.	N.U.	N.U.	N.U.
CBWLSU	N.U.	I.	N.U.	N.U.	N.U.
DOMINODL	N.U.	I.	S.I.	S.I.	I.

Please note that we compute the K-S distance and the DKW metric for all methods with respect to a reference distribution p_{ref}, as explained in 4.2.2. This reference is obtained using the following parameters, as applicable: $N_t = 1$, $K = 300$, $N_b = 30$, $N_r = 10$ and $N_u = 10$.

4.2.3.1 Influence of the number of iterations

Figures 4.5, 4.6 and 4.7 show the effect of N_t on the CV, K-S test distance (d_{ks}) and the DKW metric (d_{dkw}) for K-SVD and ODL and LRSDL. We have skipped CBWLSU and DOMINODL from this analysis because they do not accept N_t as an input. For ODL, the CV remains relatively unchanged with an increase in N_t. However, the K-SVD CV exhibits an oscillating behavior and generally high values. In case of the K-S distance, ODL shows slight increase in d_{ks} while K-SVD oscillates around a mean value that is higher than ODL. The DKW metric provides better insight: even though the ODL distributions differ from p_{ref} with increase in the iterations, the null hypothesis always holds because d_{DKW} remains positive. The d_{DKW} for K-SVD is also positive but much smaller than ODL. It also does not exhibit any specific trend with an increase in iterations. We also observed a similar behavior with the mean of similarity values. The influence of the number of iterations in LRSDL had the same oscillating behaviour as in K-SVD but with larger variation. We conclude that the number of iterations N_t does not significantly influence the metrics for both algorithms, and choose $N_t = 100$.

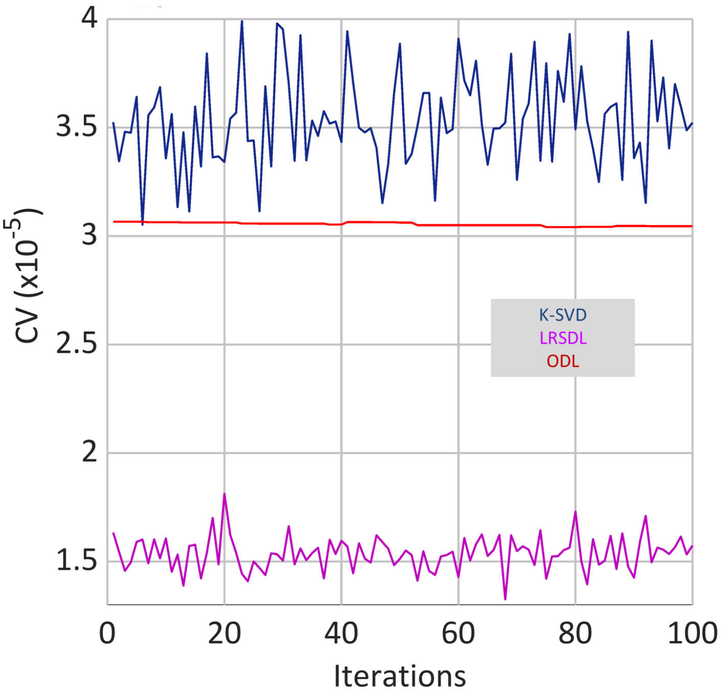


Figure 4.5 – (a) CV for K-SVD and ODL parameter analyses as a function of the number of iterations N_t.

4.2.3.2 Influence of the number of trained atoms

Figs. 4.8, 4.9 and 4.10 compare all three metrics with change in the number of trained atoms K, a parameter that is common to all DL methods. We observe that CV generally decreases with an increase in K. This indicates an improvement in the similarity between the reconstructed and the original training set. K-SVD shows an anomalous pattern for lower values of K but later converges

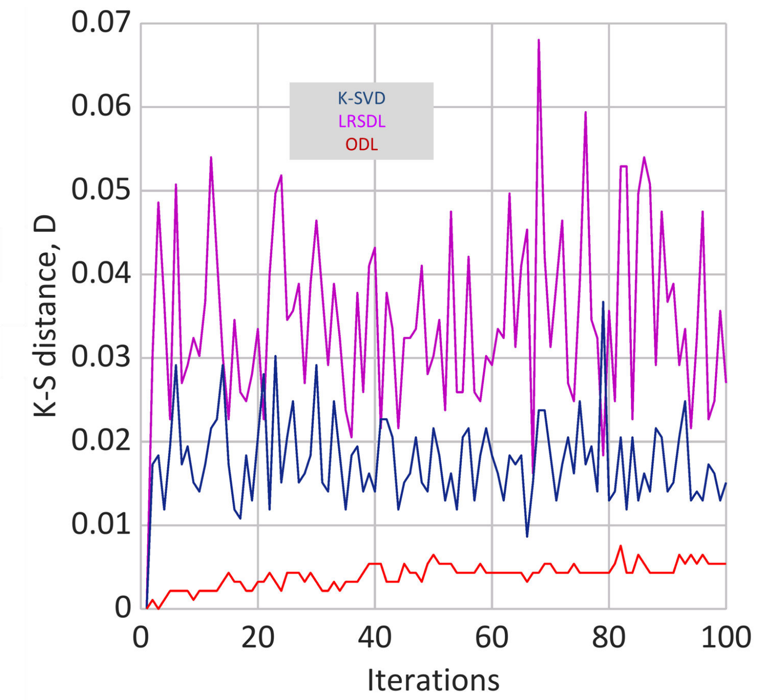


Figure 4.6 – (b) K-S distancefor K-SVD and ODL parameter analyses as a function of the number of iterations N_t.

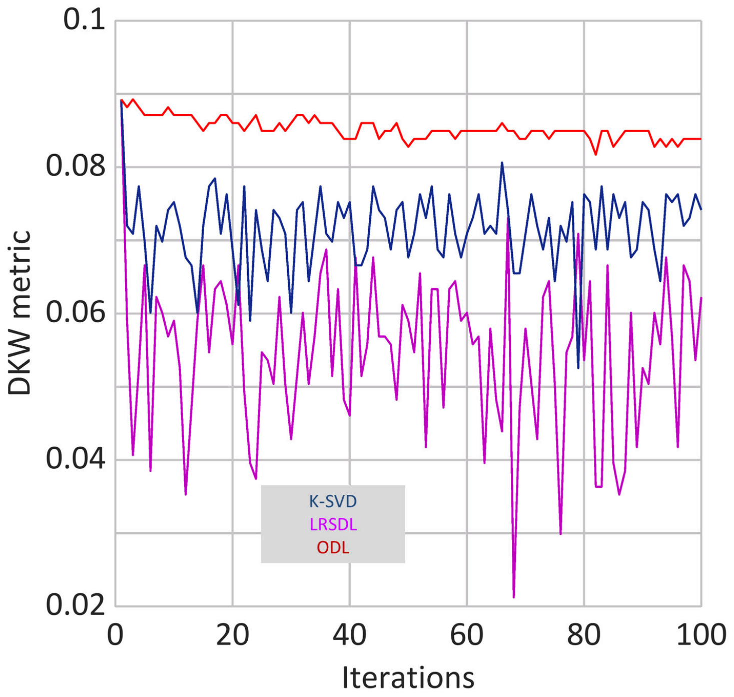


Figure 4.7 – (c) DKW metric for K-SVD and ODL parameter analyses as a function of the number of iterations N_t.

to a trend that is identical to other DL approaches. The K-S distance exhibits a linear change in the the distributions with respect to the reference. Since d_{ks} quantifies the difference between the distributions rather than stating which one is better, combining its behavior with CV makes it evident that an increase in K leads to better distributions of similarity values. The DKW metric d_{DKW}, calculated with the same reference, expectedly also shows a linear change. It is clear that, even a slight change in K leads to more negative values of d_{DKW} implying that the null hypothesis does not hold true. This shows the significant influence of the parameter K on the distributions. It was interesting to see a slight improvement for the coefficient of variation when using LRSDL with respect to the other strategies. However, KS-distance and DKW metric indicated that the distributions of similarity values for LRSDL were sensitive to the number of trained atoms only up to a certain value. The value of K is finally chosen such that the dictionary is consistently overcomplete e.g. the number of atoms is three times greater than the number of samples ($K = 640$ vs $M = 211$).

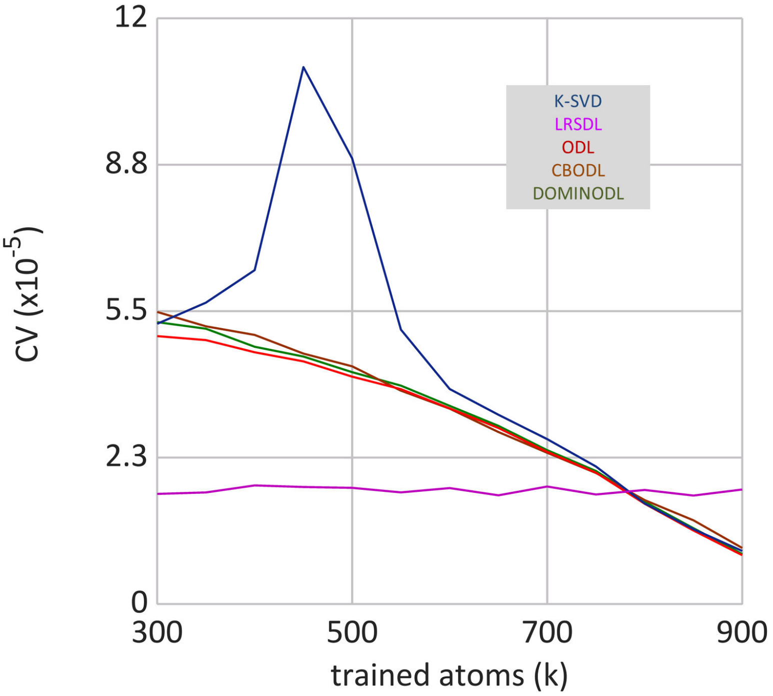


Figure 4.8 – CV for various DL algorithms as a function of the number of trained atoms K.

4.2.3.3 DOMINODL input parameters selection

It is difficult to evaluate DOMINODL EPDFs by varying all four parameters together. Instead, we fix the parameter that is common to all algorithms, i.e. the number of trained atoms K, and then determine optimal values of N_b, N_r and N_u.

Figure 4.11 shows the coefficient of variation CV of the distribution of similarity values as a function of DOMINODL parameters. The drop-off value N_u appears to have a greater influence with respect to the mini-batch dimensions N_b and N_r. To select these parameters we made considerations based on the computational time and the way the algorithm is initialized. The computational time of DOMINODL is essentially independent of N_r and N_u but slightly increases with N_b. This was verified and expected because, with N_b we are also increasing the number of SR steps (see Algorithm 4)

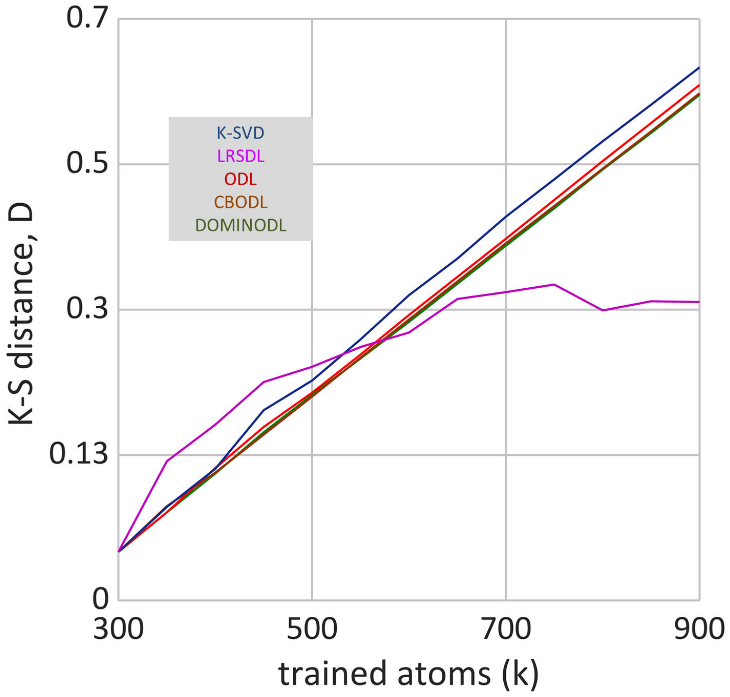


Figure 4.9 – K-S distance for various DL algorithms as a function of the number of trained atoms K.

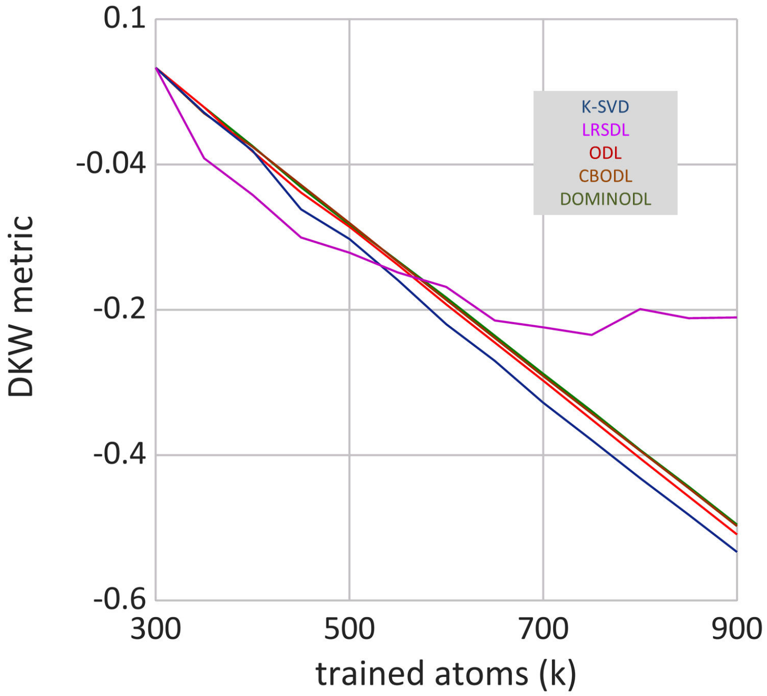


Figure 4.10 – DKW metric for various DL algorithms as a function of the number of trained atoms K.

at every iteration which is the source of bulk of computations in DL algorithms [73]. Further, in order to ensure that the correlation and the drop-off steps kick off from the very first iteration, DOMINODL should admit several new samples for each iteration thereby increasing N_b as well as the number of previous elements accordingly. Taking into account these observations, we choose $N_b = 30$ $N_r = 10$ and $N_u = 10$.

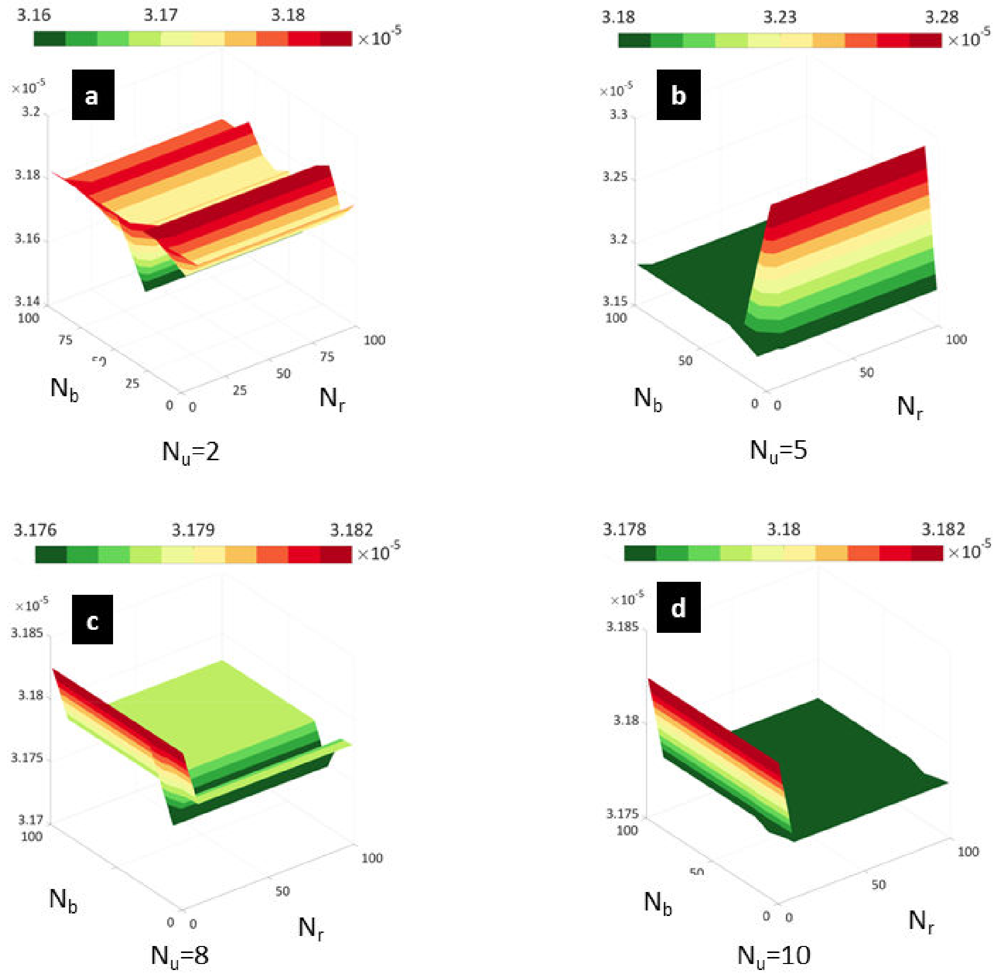


Figure 4.11 – **CV** as a function of DOMINODL input parameters for $K = 640$ and N_u as (a) 2, (b) 5, (c) 8, and (d) 10.

According to the results of the parametric evaluation, we choose the following combination of "optimal" parameters for testing our DL strategies: $\mathbf{N_t = 100}$, $\mathbf{K = 640}$, $\mathbf{N_b = 30}$, $\mathbf{N_r = 10}$, and $\mathbf{N_u = 10}$.

4.2.3.4 Considerations on computational efficiency of DL algorithms

We used a MATLAB platform on an 8-Core CPU Windows 7 desktop PC to clock the times for DL algorithms. The ODL algorithm from [35] is implemented as `mex` executable, and therefore already fine-tuned for speed. For K-SVD, we employed the efficient implementation from [72] to improve computational speed. Table 4.5 lists the execution times of the four DL approaches when using optimal input parameters. The LRSDL is the slowest of all while ODL is more than 4 times faster than K-SVD. The CBWLSU provided better classification results but is three times slower than ODL. This could be because the dictionary update step always considers the entire previous training set elements that correlate with only one new element (i.e. there is no mini-batch strategy). This makes the convergence in CBWLSU more challenging.

The DOMINODL is the fastest DL method clocking 3x speed than ODL and 15x than K-SVD. This is because the DOMINODL updates the dictionary by evaluating only a mini-batch of previous elements (instead of all of them as in CBWLSU) that correlate with a mini-batch of several new elements (CBWLSU uses just one new element). Further, DOMINODL drops out the unused elements leading to a faster convergence. We note that, unlike ODL and K-SVD implementations, we did not use `mex` executables of DOMINODL which can further shorten current execution times.From Table 4.5, the reduction in DOMINODL computational time over K-SVD is $((25.8 - 1.75) \times 100)/25.8 \approx 93\%$. The reduction for ODL and CBWLSU are computed similarly as 8% and 36%, respectively.

The computational bottleneck of mines classification lies in the training times. In comparison, the common steps of sparse decomposition and SVM-based classification during testing take just 0.4 s and 1 s, respectively, for an entire survey (1 m × 1 m area with 2500 range profiles). Thus, time taken per range profile in ca. 0.59 ms. The average scan rate of our GPR system is 0.19 m/s (or 1 cm/52.1 ms). This can go as high as 2.7 m/s (or 1 cm/3.61 ms) in other GPRs used for landmines application. Therefore, the test times do not impose much computational cost.

Table 4.5 – Computational times for DL algorithms

	DOMINODL	CBWLSU	ODL	K-SVD
Time (seconds)	1.75[1]	16.49	5.75	25.8

[1] Blue denotes the best performance among all DL algorithms

4.3 Classification results

After selecting the input parameters of the proposed DL strategies (see table 4.4), we proceed by using the obtained dictionaries for the sparse decomposition of both training and test sets. The resulting sets of sparse coefficients are the input to the SVM classifier. Figure 4.12 resumes the proposed methodology with a flow diagram, note that the labeled training set $\mathbf{Y}$ is used both for DL and classification. As mentioned in appendix B, the threshold C and the kernel function parameter γ for SVM have been selected through cross validation. Our key objective is to assess wether online DL algorithms (and in particular DOMINODL) lead to an improvement in the classification accuracy over batch learning strategies or not. As a comparison with a popular state-of-the-art classification method, we also show the classification results with a deep-learning approach based on CNN. Finally, we will show classification performances when the original samples of the range profiles are randomly reduced.

4.3.1 Classification with Optimal Parameters

For a comprehensive analysis of the classification performance, we provide both classification maps and confusion matrices for the test set $\mathbf{Y}_{\text{TEST}}$ using the optimal DL input parameters that we selected following our parametric evaluation in Section 4.2. The classification maps depict the predicted class

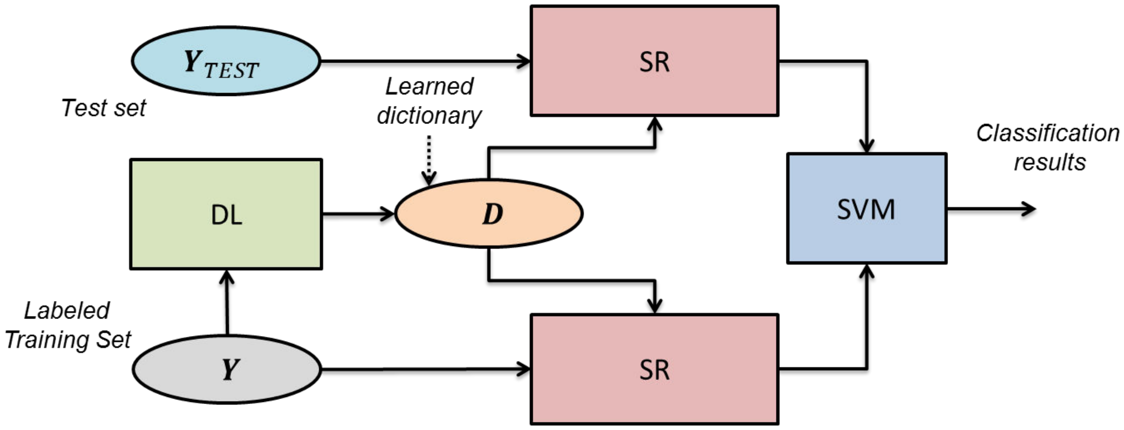


Figure 4.12 – Flow diagram which describe the proposed adaptive classification strategy

of each range profile of the survey under test. The pixel dimension of these maps is dictated by the sampling of the GPR in X and Y directions (see Table 4.1). We stacked together 3 of the 6 surveys from the test set $\mathbf{Y}_{\text{TEST}}$ where each survey had 2 buried landmines of a specific class (PMN/PMA2, ERA and Type-72).

The support of the measurements on our learned dictionaries varies from 1 to 4. Due to coherency, these few non zero elements may appear in the same position for different vector class but with slightly different values. This proved to be enough for the classifier to correctly discriminate between different classes.

Figure 4.13 shows the classification maps for different DL methods along with the raw data at depth 15 cm. The raw data in Fig. 4.13(a) shows that only four of the six mines exhibit a strong reflectivity while the other two mines have echoes so weak that they are not clearly visible in the raw data. Figures 4.13(b)-(d) show the results of the SR-based classification approaches using DL. All methods clearly detect and correctly classify the large PMN/PMA2 mines. In case of the medium-size ERA, the echoes are certainly detected as non-clutter but some of its constituent pixels are incorrectly classified as another mine. It is remarkable that the left ERA mine is recognized by our method even though it cannot be discerned visually in the raw data. Most of the false alarms in the map belong to the smallest Type-72 mines. This is expected because their small sizes produce echoes very similar to the ground clutter. On the other hand, when T-72 is the ground truth, it is correctly identified.

Using accurate ground truth information, we defined *target halos* as the boundaries of the buried landmines. The dimension of the target halos varied depending on the mine size. Let the number of pixels and the declared mine pixels inside the target halo (for a certain class) be n_t and n_m, respectively. Similarly, we denote the number of true and declared clutter pixels outside the target halo by n_c and n_d, respectively. Then, the probabilities of correct classification (P_{CC}) for each target class and clutter are, respectively,

$$P_{CC_{\text{mines}}} = \frac{n_m}{n_t}, \text{ and } P_{CC_{\text{clutter}}} = \frac{n_d}{n_c}. \tag{4.15}$$

The P_{CC} being the output of a classifier should not be mistaken as the radar's probability of detection P_d which is the result of a detector. A detector declares the presence of a mine when only a few pixels inside the halo have been declared as mine. P_{CC} provides a fairer and more accurate evaluation of the classification result. This per-pixel information can be easily used to improve the final detec-

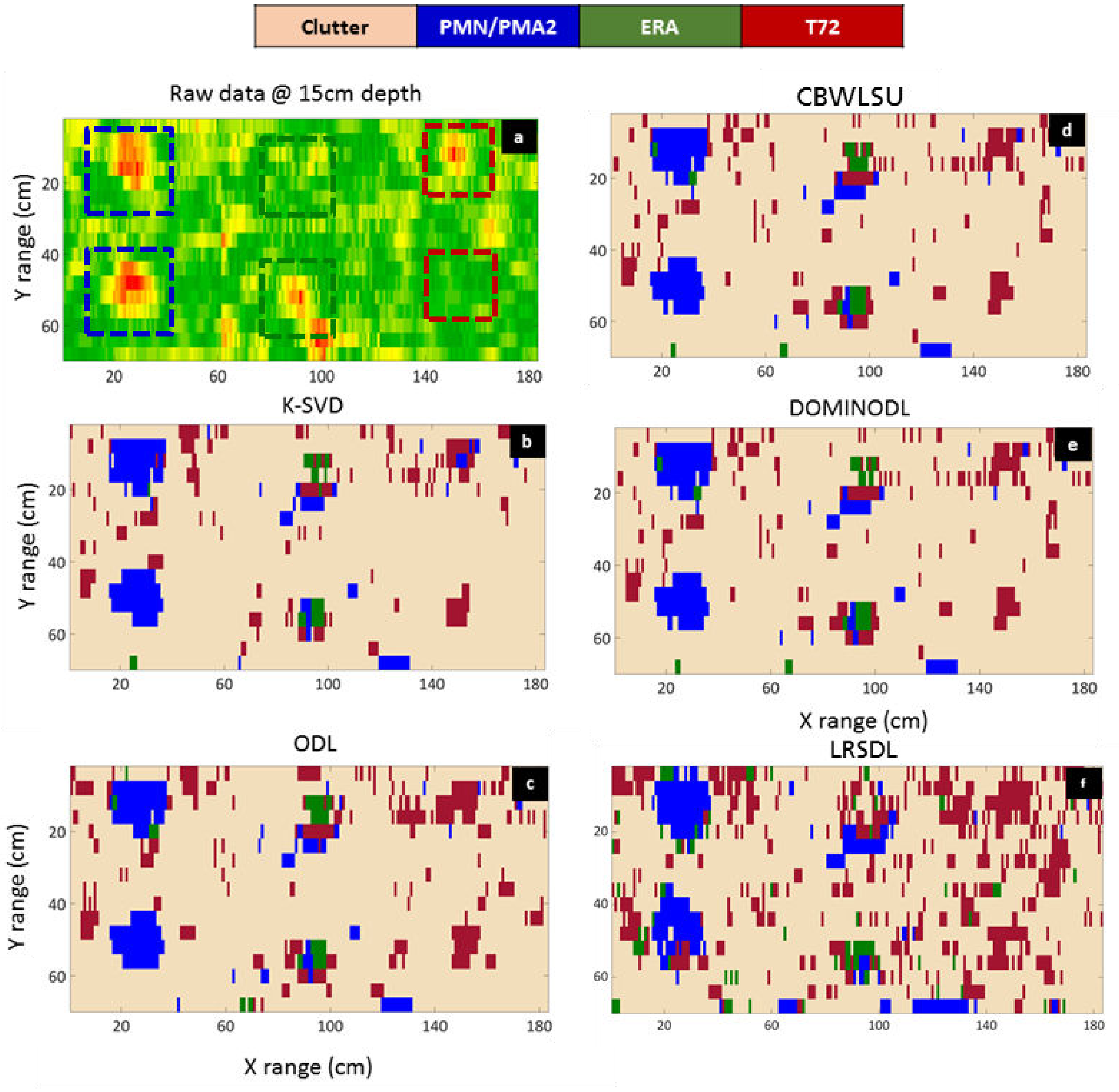


Figure 4.13 – (a) Raw data at 15 cm depth. The classification maps of the same area containing 6 buried landmines using an SR-based approach with dictionary learned using (b) K-SVD, (c) ODL, (d) CBWLSU, (e) DOMINODL and (f) LRSDL algorithms with optimally selected input parameters.

tion result. For instance, the operator could set a threshold for the minimum number of pixels to be detected in a cluster so that a circle with center at the cluster centroid could be used as the detected mine. However, such a circle may exclude some of the mine pixels leading to a potential field danger. The per-pixel classification is then employed to determine the guard area around the mine circle.

A *confusion matrix* is a quantitative representation of the classifier performance. The matrix lists the probability of classifying the ground truth as a particular class. The classes listed column-wise in the confusion matrix are the ground truths while the row-wise classes are their predicted labels. Therefore, the diagonal of the matrix is the P_{CC} while off-diagonal elements are probabilities of misclassification. For the classification map of Fig. 4.13, table 4.6 shows the corresponding confusion matrices for each DL-based classification approach. In general, we observe an excellent classification

Table 4.6 – Confusion matrix with optimal DL input parameter selection.

		Clutter	PMN/PMA2	ERA	Type-72
K-SVD	Clutter	0.892	0.044	0.25	0.37
	PMN/PMA2	0.022	0.938[1]	0.166	0.074
	ERA	0.021	0.017	0.472	0.018
	Type-72	0.064	0	0.111	0.537
LRSDL (SRC)	Clutter	0.435	0.061	0.111	0.351
	PMN/PMA2	0.155	0.289	0.319	0.259
	ERA	0.172	0.372	0.361	0.278
	Type-72	0.237	0.272	0.208	0.111
ODL	Clutter	0.871	0	0.194	0.333
	PMN/PMA2	0.022	0.973	0.139	0
	ERA	0.018	0.026	0.583	0.018
	Type-72	0.088	0	0.083	0.648
CBWLSU	Clutter	0.872	0.017	0.181	0.314
	PMN/PMA2	0.023	0.973	0.153	0
	ERA	0.025	0.008	0.528	0
	Type-72	0.08	0	0.138	0.685
DOMINODL	Clutter	0.876	0.017	0.167	0.315
	PMN/PMA2	0.023	0.974	0.138	0
	ERA	0.027	0.008	0.58	0
	Type-72	0.077	0	0.11	0.685

[1] Gray denotes the $\mathbf{P}_{\mathbf{CC}}$ value for a specified class and DL algorithm

of PMN/PMA2 landmines ($\sim$ 98%), implying that almost every range profile in the test set which belongs to this class is correctly labeled. The P_{cc} for the clutter is also quite high ($\sim$ 90%). This can also be concluded from the classification maps where the false alarms within the actual clutter regions are very sparse (i.e. they do not form a cluster) and, therefore, unlikely to be interpreted as an extended target. As noted previously, most of the clutter misclassification is associated with the Type-72 class. The ERA test targets show some difficulties with correct classification. However, most of the pixels within its target halo are declared at least as some type of mine (which is quite useful in terms of issuing safety warnings in the specific field area). This result can be explained by the fact that ERA test targets (being simulant landmines, i.e. SIMs) do not represent a specific mine but have general characteristics common to most landmines. The Type-72 mines exhibit a P_{cc} which is slightly higher with respect to ERA targets. This is a remarkable result because Type-72 targets were expected to be the most challenging to classify due to their small size.

Conventionally, as mentioned in [88], LRSDL is associated with a sparse-representation-based classification (SRC) technique. However, applying this approach to our problem resulted in very low accuracy (an average of ca. 20% across all classes as evident from Table 4.6) and semi-random classification maps (Fig. 4.13). This can be explained by the extreme similarity between the training set examples of different classes; mines and clutter are only slightly dissimilar in their responses and mine responses are generally hidden in the ground reflections. Each learned "block" D_c differed only slightly from the other and, therefore, poor classification results are achieved with this dataset. LRSDL won't be used for further evaluations.

All DL algorithms used for our sparse classification approach show very similar results for the clutter and PMN/PMA2 classes. However, online DL methods show higher P_{CC} for the ERA and Type-72 targets with respect to K-SVD.

From Table 4.6, the detection enhancement using the best of the online DL algorithms for PMN/PMA2 over K-SVD is $((0.974 - 0.938) \times 100)/0.938 \approx 4\%$. The improvements for ERA and T-72 are computed similarly as 23% and 28%, respectively.

4.3.2 Classification with Non-Optimal Parameters

In order to demonstrate how the quality of the learned dictionary affects the final classification, we now show the confusion matrices for a non-optimal selection of input parameters in different DL algorithms. Our goal is to emphasize the importance of learning a good dictionary by selecting the optimal parameters rather than specifying how each parameter affects the final classification result. We arbitrarily selected the number of trained atoms K to be only 300 for all DL approaches, reduce the number of iterations to 25 for ODL and KSVD and, for DOMINODL, we use N_r=30, N_b=5 and N_u=2. Table 4.7 shows the resulting confusion matrix. While the clutter classification accuracy is almost the same as in Table 4.6, the P_{cc} for PMN/PMA2 landmines decreased by $\sim$ 10% for most of the algorithms except ODL where it remains unchanged. The classification accuracy for ERA and Type-72 mines is only slightly worse for online DL approaches. However, in the case of K-SVD, the P_{CC} reduces by $\sim$ 30% and $\sim$ 10% for ERA and Type-72, respectively. Clearly, the reconstruction and correct classification of range profiles using batch algorithms such as K-SVD is strongly affected by a non-optimal choice of DL input parameters. As discussed earlier in Section 4.2.3, this degradation is likely due to the influence of K rather than N_t.

4.3.3 Comparison with Deep Learning Classification

The core idea of SR-based classification is largely based on the assumption that signals are linear combinations of a few atoms. In practice, this is often not the case. This has led to a few recent works [132] that suggest employing deep learning for radar target classification. However, these techniques require significantly large datasets for training. We compared classification results of our methods with a deep learning approach. In particular, we constructed a CNN because these networks are known to efficiently exploit structural or locational information in the data and yield comparable learning potential with far fewer parameters [133]. We modeled our proposed CNN framework as a classification problem wherein each class denotes the type of mine or clutter. The training data set for

Table 4.7 – Confusion matrix with non-optimal DL input parameter selection

		Clutter	PMN/PMA2	ERA	Type-72
K-SVD	Clutter	0.853	0.07	0.305	0.222
	PMN/PMA2	0.037	0.851	0.222	0.111
	ERA	0.032	0	0.194	0.241
	Type-72	0.077	0.078	0.277	0.426
ODL	Clutter	0.86	0.017	0.181	0.444
	PMN/PMA2	0.016	0.973	0.097	0
	ERA	0.022	0.008	0.638	0
	Type-72	0.1	0	0.083	0.555
CBWLSU	Clutter	0.887	0.078	0.319	0.352
	PMN/PMA2	0.019	0.877	0.097	0
	ERA	0.018	0.043	0.541	0
	Type-72	0.074	0	0.042	0.648
DOMINODL	Clutter	0.888	0.078	0.319	0.352
	PMN/PMA2	0.019	0.877	0.097	0
	ERA	0.018	0.043	0.54	0
	Type-72	0.074	0	0.042	0.648

our CNN structure is the matrix **Y** (see 4.1.3). Building up a synthetic database is usually an option for creating (or extending) a training set for deep learning applications. However, accurately modeling a GPR scenario is still an ongoing challenge in the GPR community because of the difficulties in accurately reproducing the soil inhomogeneities (and variabilities), the surface and underground clutter, the antenna coupling and ringing effects, etc. Even though some applications have been promising [134], this remains a cumbersome task. The input layer of our CNN took one-dimensional range profiles with 211 samples. It was followed by two convolutional layers with 20 and 5 filters of size 20 and 10, respectively. The output layer consisted of four units wherein the network classifies the given input data as clutter or one of the three mines. There were rectified linear units (ReLU) after each convolutional layer; the ReLU function is given by $\mathrm{ReLU}(x) = \max(x, 0)$ [135].

The architecture of the CNN was selected through an arduous process of testing many combination of layers/filters and hyperparameters which would lead to better accuracy during training. A deeper network slightly increased the accuracy in the training phase but led to poorer performance when classifying new data (i.e. the test set $\mathbf{Y}_{test}$). Since our data are limited, adding more layers (i.e. more weights) only led to overfitting and made the network incapable to generalize on new datasets. A multi-dimensional CNN formed by clustering 2D and 3D data would have further reduced the training set. Augmenting the data was also envisioned but commonly used transformations such as scaling/rotations are not useful in our case because the mines were always in the same inclination and their dimension defines the class itself. We also attempted adding different levels of noise but this did not lead to better results considering the available data are already very noisy.

We trained the network with the labeled training set **Y**, selecting $\sim$ 20% of the training data for validation. Specifically, the validation set employed 100, 25, 25, and 25 range profiles for clutter, PMN/PMA2, ERA and Type-72, respectively. We used a stochastic gradient descent algorithm for updating the network parameters with the learning rate of 0.001 and mini-batch size of 20 samples for 2000 epochs. We realized the proposed network in TensorFlow on a Windows 7 PC with 8-core CPU. The network training took 3.88 minutes. Figure 4.14 shows the classification map obtained using CNN. The corresponding confusion matrix is listed in Table 4.8. We note that the CNN classifier shows worse P_{CC} than our SR-based techniques, particularly for ERA and Type-72 target classes.

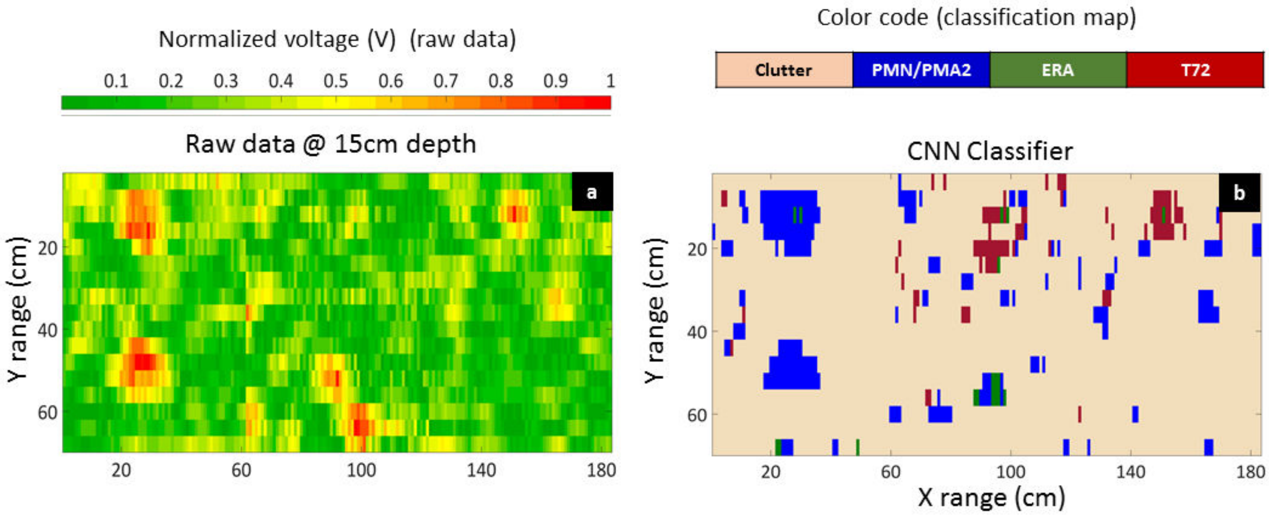


Figure 4.14 – (a) Raw data at 15 cm depth. (b) Classification maps of the same area containing 6 buried landmines using CNN-based classification.

Table 4.8 – Confusion matrix for CNN-based classification

	Clutter	PMN/PMA2	ERA	Type-72
Clutter	0.909	0.14	0.38	0.574
PMN/PMA2	0.036	0.807	0.181	0
ERA	0.022	0.053	0.319	0.056
Type-72	0.033	0	0.111	0.370

4.3.4 Classification with Reduced Range Samples

We now analyze the robustness of our DL-based adaptive classification method to the reduction of the number of samples in the raw data. Assuming the collected data $\mathbf{Y}_{\text{TEST}}$ is sparse in dictionary $\mathbf{D}$, we undersampled the original raw data $\mathbf{Y}_{\text{TEST}}$ in range to obtain its row-undersampled version $\widetilde{\mathbf{Y}}_{\text{TEST}}$ by randomly reducing the samples. We then applied the same random sampling pattern to the dictionary $\mathbf{D}$ for obtaining the sparse coefficients. We also analyzed the CNN classifier when the signals are randomly reduced in the same way. Figure 4.15 illustrates the classification map for all DL approaches when the sampling is reduced by 50%. Tables 4.9,4.10 and4.11 show confusion matrices when undersampling by 25%, 50%, and 75%.

In comparison to the results in Table 4.6 which used all samples of the raw data, the DL approaches maintain similar classifier performance even when we reduce the samples by 75% (i.e. just 52 samples in total). In contrast, the CNN classifier result which is already heavily compromised with a reduction of 25%, fails completely for 50%and 75% sampling rate.

Reducing the number of signal samples when using a dictionary which minimizes the number of non-zero entries in the sparse representation, still assures an exact reconstruction of the signal itself and, consequently its correct classification. The features for classifying the traces are thus robust to the reduction of the original samples. Deep learning strategies use the signal samples directly as classification features. They also require enormous amount of data for training. Therefore, the degradation in their performance is expected. From the confusion matrix in Table 4.11 indicates that CNN has the highest P_{CC} for ERA. This is a false trail because the network mis-classified almost every pixel as ERA. Overall, DOMINODL and CBWLSU provide excellent results for small mines. However, as seen earlier, CBWLSU is not very well-suited for real-time operation because of longer execution times.

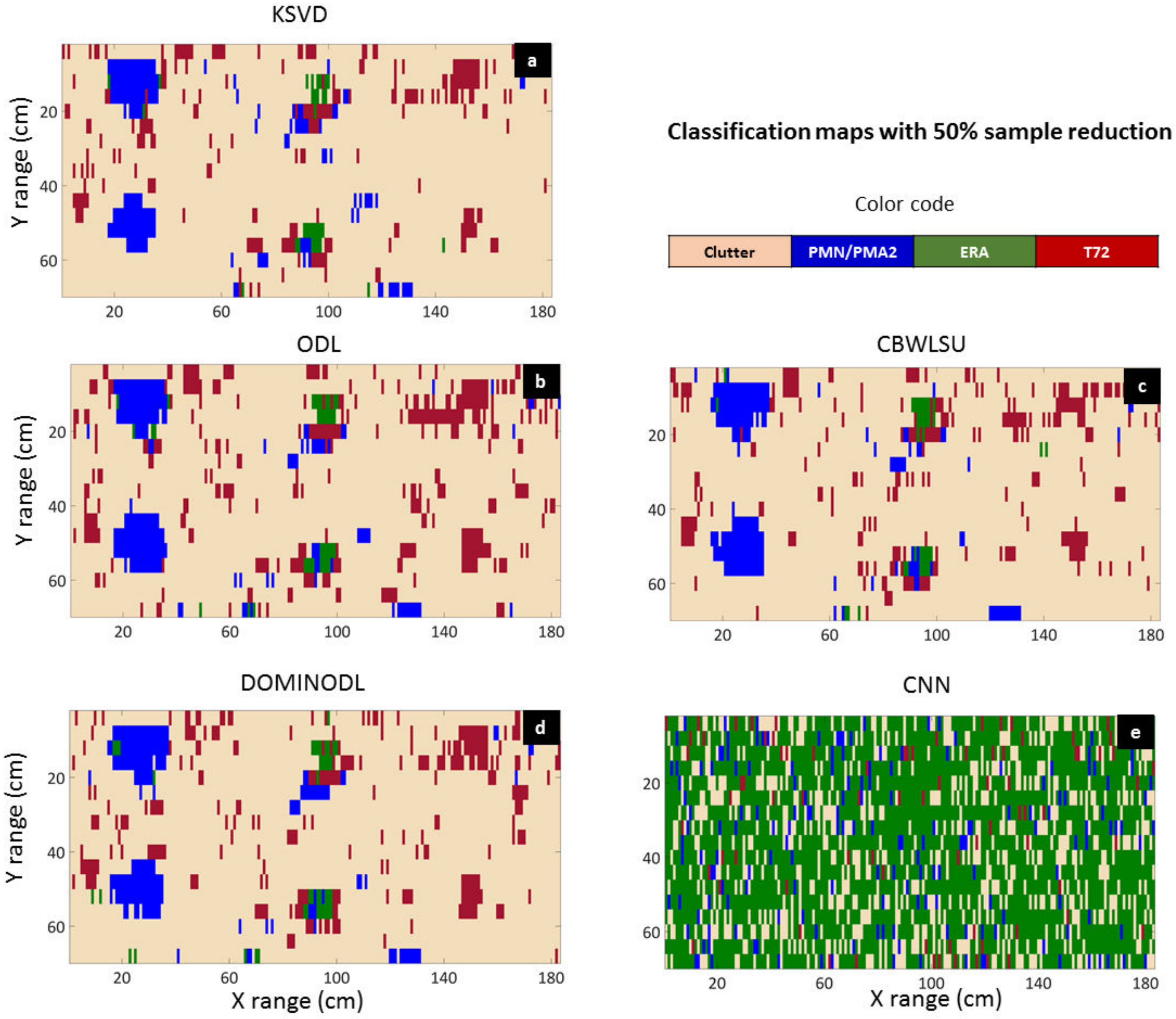


Figure 4.15 – The classification maps of the same area containing 6 buried landmines using an SR-based approach with dictionary learned using (a) K-SVD, (b) ODL, (c) CB-WLSU, and (d) DOMINODL algorithms. The input parameters were optimally selected and the number of samples were reduced by 50%. (e) The corresponding result with reduced samples for CNN-based classification.

Table 4.9 – Confusion matrices for different DL algorithms and CNN with 25% samples reduction

		Clutter	PMN/PMA2	ERA	Type-72
K-SVD	Clutter	0.892	0.078	0.319	0.389
	PMN/PMA2	0.021	0.921	0.153	0.055
	ERA	0.021	0	0.486	0.018
	Type-72	0.065	0	0.041	0.537
ODL	Clutter	0.872	0.088	0.208	0.315
	PMN/PMA2	0.024	0.956	0.139	0
	ERA	0.018	0.017	0.527	0.018
	Type-72	0.087	0	0.111	0.666
CBWLSU	Clutter	0.871	0.026	0.194	0.351
	PMN/PMA2	0.024	0.956	0.139	0
	ERA	0.025	0.017	0.541	0
	Type-72	0.79	0	0.125	0.648
DOMINODL	Clutter	0.88	0.017	0.236	0.277
	PMN/PMA2	0.022	0.964	0.138	0
	ERA	0.018	0.017	0.527	0
	Type-72	0.078	0	0.097	0.722
CNN	Clutter	0.708	0.359	0.333	0.407
	PMN/PMA2	0.026	0.41	0.097	0.018
	ERA	0.236	0.21	0.5	0.426
	Type-72	0.029	0.017	0.069	0.148

Table 4.10 – Confusion matrices for different DL algorithms and CNN with 50% samples reduction

		Clutter	PMN/PMA2	ERA	Type-72
K-SVD	Clutter	0.882	0.026	0.291	0.37
	PMN/PMA2	0.018	0.947	0.153	0.037
	ERA	0.021	0.026	0.5	0
	Type-72	0.078	0	0.055	0.592
ODL	Clutter	0.868	0	0.208	0.333
	PMN/PMA2	0.021	0.965	0.18	0.018
	ERA	0.018	0.035	0.5	0
	Type-72	0.09	0	0.111	0.648
CBWLSU	Clutter	0.872	0.017	0.25	0.40
	PMN/PMA2	0.023	0.973	0.111	0
	ERA	0.02	0.008	0.541	0
	Type-72	0.083	0	0.097	0.592
DOMINODL	Clutter	0.868	0.035	0.194	0.296
	PMN/PMA2	0.023	0.929	0.138	0
	ERA	0.024	0.035	0.527	0.018
	Type-72	0.083	0	0.138	0.685
CNN	Clutter	0.265	0.166	0.181	0.148
	PMN/PMA2	0.062	0.096	0.069	0.018
	ERA	0.645	0.728	0.736	0.75
	Type-72	0.027	0.088	0.014	0.074

Table 4.11 – Confusion matrices for different DL algorithms and CNN with 75% samples reduction

		Clutter	PMN/PMA2	ERA	Type-72
K-SVD	Clutter	0.877	0.061	0.402	0.426
	PMN/PMA2	0.02	0.912	0.125	0.074
	ERA	0.021	0.026	0.333	0.018
	Type-72	0.08	0	0.138	0.481
ODL	Clutter	0.862	0.02	0.319	0.296
	PMN/PMA2	0.023	0.964	0.138	0.018
	ERA	0.021	0.008	0.416	0.074
	Type-72	0.091	0	0.125	0.611
CBWLSU	Clutter	0.855	0.088	0.388	0.370
	PMN/PMA2	0.023	0.973	0.111	0
	ERA	0.027	0.017	0.333	0.018
	Type-72	0.091	0	0.125	0.611
DOMINODL	Clutter	0.864	0.035	0.278	0.444
	PMN/PMA2	0.027	0.938	0.152	0
	ERA	0.026	0.026	0.5	0
	Type-72	0.082	0	0.069	0.556
CNN	Clutter	0.162	0.105	0.152	0.129
	PMN/PMA2	0.015	0.061	0.013	0
	ERA	0.647	0.71	0.708	0.759
	Type-72	0.17	0.12	0.12	0.11

Chapter 5

Summary and conclusions

5.1 Existing challenges and proposed methodology

APM detection/classification with UWB GPR is an extremely difficult task. Undesired effects coming from antenna ringing, strong clutter, inhomogeneous soil and surface roughness contribute to distort the backscattered signals from the targets of interest. Moreover, APMs have a very weak signal response due to their low permittivity contrast w.r.t. the soil. Various classification approaches rely on databases of synthetic (or measured) signatures to discriminate between mines and clutter, however, to create a general database which takes into account all possible GPR scenario parameters and inhomogeneities is a cumbersome task. An alternative is to extract discriminative features from a representative database which only includes the targets of interest.

In this work, our main goal was to extract relevant features from real GPR measurements of buried APMs via sparse representation (SR) and exploit them for classification. In order that the SR will efficiently represent the measurements, one must obtain an appropriate *dictionary*. The core of our approach is to evaluate different dictionary learning (DL) strategies, using a representative set of measurements (i.e. a *training set*) to *learn* the dictionary which will yeld characteristics features for the classification of our GPR dataset.

In particular, we compare online-DL with batch-DL strategies and use the obtained sparse coefficients as input features of a Support Vector Machine Classifier which discriminates between different types of mines and clutter. Among the tested DL strategies, we also introduce our novel DOMINODL Online-DL algorithm which is able to learn faster than other DL methods and avoid redundant use of the the training set by analyzing it in mini-batches. The employed DL algorithms are K-SVD, LRSDL, ODL, CBWLSU and the proposed DOMINODL

5.2 Evaluation of DL algorithms

Our classification approach is sensitive to the input parameters of DL algorithms such as the dimension of the training set or the number of iterations. In 4.2 we propose a method to investigate the effect of these parameters on the learning performance in order to select optimal parameter values that yield a dictionary which is optimized to sparsely represent our GPR data.

We use a similarity measure based on the cross correlation formula to quantify the closeness of the original training set $\mathbf{Y}$ with the reconstructed one $\hat{\mathbf{Y}}$, the latter being obtained by sparse reconstruction with the learned dictionary. We generate Empirical Probability Density Functions (EPDFs) of similarity measures for every proposed DL algorithms using a particular combination of input parameters and analyze these EPDFs (and Cumulative Distribution Density Functions ECDFs) using statistical metrics such as the coefficient of variation (CV), the 2-sample Kologorov Smirnoff-Distance (d_{ks}) and the Dvoretzky-Kiefer-Wolfowitz inequality metric (d_{dkw}). The latter two, being based on a distance calculation, are computed according to a reference distribution.

We noticed that increasing the number of iterations (N_t) generates distributions with slightly better metrics for ODL while K-SVD and LRSDL have an oscillating trend; with LRSDL having the lower CV. However, the null hypotesis for the DKW metric always stays true for all of them meaning that the change in the distributions is marginal. ODL, K-SVD and LRSDL are the only algorithms which used N_t as an input.

The number of learned atoms (K) has a significant impact on all DL algorithms. Increasing K leads to a smaller CV for all algorithms apart from LRSDL, where CV is almost constant w.r.t. K and lower than for the others algorithms. The null hypotesis for d_{dkw} fails very quickly meaning that distributions tend to change a lot for different values of K and produce EPDFs which are more skewed to the unity value of normalized frequencies. LRSDL has a different behaviour since its d_{dkw} tends to stay constant after a certain number of training atoms indicating no particular benefits/change after a certain increase in K

A dedicated evaluation of the DOMINODL input parameters (N_b,N_r, and N_u, being the dimension of the mini-batch of new elements, of previous elements and the drop-off value) showed that N_u appears to have a greater influence w.r.t. N_b and N_r, however it is important to select N_b and N_r so that the correlation steps start from the first iterations and the value of N_u so that the drop off happens during the iterations.

The DOMINODL algorithm is the fastest DL method (3x speed than ODL and 15x than K-SVD), LRSDL is the slowest needing ca. 17 minutes to train the dictionary.

5.3 Classification performances

In 4.3 we show the results of our DL-based classification approach on experimental GPR measurements with buried landmine simulants, along with some additional comparison/tests. The classification of the SR coefficients obtained with the different DL algorithms has been performed with a Support Vector Machine classifier (SVM) using a Radial Basis Function (RBF) kernel. When using optimal input parameters for the DL algorithms (obtained with the aforementioned evaluation) we observe an excellent classification of PMN/PMA2 landmines ($P_{cc} = 98\%$), and clutter ($P_{cc} = 90\%$). The classification maps show very sparse false alarms inside the clutter region which is unlikely to be mistaken with targets.Most of the clutter mis-classification is associated with the Type-72 class. ERA test targets are not always correctly labeledbut always detected as a threat. The P_{cc} is around 58% while, if we sum up wrong target declarations inside the ERA regions, we obtain a *detection rate* above 80%. The Type-72 classification accuracy is higher with respect to ERA targets, despite being the smallest mines in the set ($P_{cc} = 68.5\%$). All DL algorithms used in our sparse classification approach achieve very similar results for the clutter and PMN/PMA2 classes. However, online DL methods show higher P_{CC} for the ERA and Type-72 targets with respect to K-SVD and LRSDL. For the latter, due to the class-discriminative nature of the obtained dictionary, the classification is residual-based (see [40]) resulting in a very poor accuracy. Therefore LRSDL was not used for further classification tests.

Using non-optimal parameter values for the proposed DL algorithms leads to worse classification performance, specially for K-SVD, demonstrating how important is to properly select DL input parameters for the training set at hand and how online-DL algorithms are more robust to a non-optimal selection.

As an additional comparison with a state-of-the art classification algorithm, we use a Convolutional Neural Network (CNN) trained with the same training set used for our DL-based approach. Classification results with CNN are poorer for the classification of clutter and landmines, particularly ERA and T72 P_{cc} with an accuracy that degrades more than 27%. Using a deeper structure increased slightly the accuracy in the training phase but led to poorer performance when classifying new data (i.e. the test set $\mathbf{Y}_{test}$). Since the amount of experimental data is limited, adding more layers (i.e. more weights) only led to overfitting and made the network incapable of generalize on new datasets. We also tested the implementation of a 3D CNN, by re-organizing the training sets in 3D clusters. However, we ended up with much less data examples (only 68 examples for training/test). The problem of overfitting was more prominent and, in order for the network to yield any significant outcome, we had to use only a single convolutional layer. The network was able to recognize some patterns of mines (especially the bigger ones) but failed completely in the class declaration.

Our final evaluation consisted in assessing the classification performances when reducing the original samples of the GPR range profiles to 25% 50% and 75%. This can be seen as a connection with the classical Compressive Sensing applications where one uses incomplete (or compressed) measurements to reconstruct the original signal. In our case we wanted to see how this reduction affects the P_{CC} of mines and clutter. CNN reduces drastically its classification accuracy even when the reduction is only 25% and completely fails when increasing the rate of undersampling. DL-based approach performances are definitely more robust to the sample reduction with mixed results depending on the type of mines and rate of undersampling. As an example, classification accuracy using DOMINODL drops only by 1.2% for clutter, 3.5% for PMN/PMA2, 8% for ERA and 12.9% for T72. Here, we even obtained the dictionary on the full profiles and classify with the reduced ones.

5.4 Final remarks and outlook

The evaluation of the proposed DL algorithms for our experimental GPR dataset with buried landmine simulants showed that Online-DL algorithms can provide better reconstruction performances than their batch counterparts (specially for smaller APMs) while being significantly faster. DOMINODL is the fastest one and employs only correlated mini-batches of previous and new training set elements, dropping off unused ones.

Our approach is able to detect 4 different classes of APM responses, separating them well from clutter; most of the classes are also correctly classified. DL approaches are robust to the reduction of signal samples, outperforming a simple CNN classifier.

This approach is not limited to GPR datasets and can be applied to other classification problems and data types. Due to the extremely fast learning time of Online-DL strategies (specially DOMINODL), real time learning of a constantly varying training set (updating it with new measurements) it is also envisioned, paving the way for a fully adaptive classification approach.

Bibliography

[1] L. Monitor, B. Policy, and M. Action, "Landmine monitor 2013," *Concord: International Campaign to Ban Landmines*, 2013.

[2] M. A. González-Huici, "Accurate ground penetrating radar numerical modeling for automatic detection and recognition of antipersonnel landmines," Ph.D. dissertation, Universitäts-und Landesbibliothek Bonn, 2013.

[3] L. Robledo, M. Carrasco, and D. Mery, "A survey of land mine detection technology," *International Journal of Remote Sensing*, vol. 30, no. 9, pp. 2399–2410, 2009.

[4] D. J. Daniels, *Ground penetrating radar*. IET, 2004.

[5] H. M. Jol, Ed., *Ground penetrating radar theory and applications*. Elsevier Science, 2009.

[6] R. Persico, *Introduction to ground penetrating radar: Inverse scattering and data processing*. John Wiley & Sons, 2014.

[7] A. G. Yarovoy, P. van Genderen, and L. P. Ligthart, "Ultra-wideband ground penetrating impulse radar," in *Ultra-wideband, short-pulse electromagnetics 5*, P. D. Smith and S. R. Cloude, Eds. Springer, 2002, pp. 183–189.

[8] F. Giovanneschi, M. A. González-Huici, and U. Uschkerat, "A parametric analysis of time and frequency domain GPR scattering signatures from buried landmine-like targets," in *SPIE Defense, Security, and Sensing*, 2013, pp. 870 914–870 914.

[9] C. Bruschini and B. Gros, "A survey of research on sensor technology for landmine detection," *J. Conventional Weapons Destruction*, vol. 2, no. 1, p. 3, 2016.

[10] C. R. Ratto, P. A. Torrione, and L. M. Collins, "Exploiting ground-penetrating radar phenomenology in a context-dependent framework for landmine detection and discrimination," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 5, pp. 1689–1700, 2011.

[11] M. A. González-Huici and F. Giovanneschi, "A combined strategy for landmine detection and identification using synthetic GPR responses," *J. Appl. Geophys.*, vol. 99, pp. 154–165, 2013.

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

[13] I. Giannakis, A. Giannopoulos, and A. Yarovoy, "Model-based evaluation of signal-to-clutter ratio for landmine detection using ground-penetrating radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3564–3573, 2016.

[14] A. B. Suksmono, E. Bharata, A. A. Lestari, A. G. Yarovoy, and L. P., "Compressive stepped-frequency continuous-wave ground penetrating radar," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 665–669, 2010.

[15] Y. C. Eldar and G. Kutyniok, *Compressed sensing: Theory and applications*. Cambridge University Press, 2012.

[16] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[17] E. J. Candes and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Computational Mathematics*, vol. 6, no. 2, pp. 227–254, 2006.

[18] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[19] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies," *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[20] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] I. Chant, D. Lee, and D. Ireland, "Dsto landmine detection test targets," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS SCIENCES LAB, Tech. Rep., 2005.

[22] B. G. R. De Prony, "Essai éxperimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool,a différentes températures," *Journal de l'école polytechnique*, vol. 1, no. 22, pp. 24–76, 1795.

[23] C. Carathéodory, "Über den variabilitätsbereich der fourier'schen konstanten von positiven harmonischen funktionen," *Rendiconti Del Circolo Matematico di Palermo (1884-1940)*, vol. 32, no. 1, pp. 193–217, 1911.

[24] I. F. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with focuss: a recursive weighted minimum norm algorithm," *Electroencephalography and clinical Neurophysiology*, vol. 95, no. 4, pp. 231–251, 1995.

[25] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[26] A. Beurling, "Sur les intégrales de fourier absolument convergentes et leur application à une transformation fonctionelle," in *Ninth Scandinavian Mathematical Congress*, 1938, pp. 345–366.

[27] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[28] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.

[29] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[30] E. Van Den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[31] E. van den Berg and M. P. Friedlander, "Spgl1: A solver for large-scale sparse reconstruction," 2007.

[32] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[33] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 5, 1999, pp. 2443–2446.

[34] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.

[36] J. H. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, 2010.

[37] E. Baransky, G. Itzhak, N. Wagner, I. Shmuel, E. Shoshan, and Y. Eldar, "Sub-nyquist radar prototype: Hardware and algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 2, pp. 809–822, 2014.

[38] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank, "Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, 2009.

[39] A. Correas-Serrano and M. A. González-Huici, "Experimental evaluation of compressive sensing for doa estimation in automotive radar," in *2018 19th International Radar Symposium (IRS)*. IEEE, 2018, pp. 1–10.

[40] F. Giovanneschi and M. A. González-Huici, "A preliminary analysis of a sparse reconstruction based classification method applied to GPR data," in *Int. Workshop Advanced Ground Penetrating Radar*, 2015, pp. 1–4.

[41] W. Shao, A. Bouzerdoum, and S. L. Phung, "Sparse representation of gpr traces with application to signal classification," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 7, pp. 3922–3930, 2013.

[42] F. Giovanneschi, K. V. Mishra, M. A. Gonzalez-Huici, Y. C. Eldar, and J. H. Ender, "Dictionary learning for adaptive gpr landmine classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 036–10 055, 2019.

[43] J. L. Davis and A. Annan, "Ground-penetrating radar for high-resolution mapping of soil and rock stratigraphy 1," *Geophysical prospecting*, vol. 37, no. 5, pp. 531–551, 1989.

[44] P. Annan, "Ground penetrating radar principles, procedures and applications," *Sensors and software*, vol. 278, 2003.

[45] R. J. Knight and A. L. Endres, "An introduction to rock physics principles for near-surface geophysics," in *Near-surface geophysics*. Society of Exploration Geophysicists, 2005, pp. 31–70.

[46] J. Skaar, "Fresnel equations and the refractive index of active media," *Physical Review E*, vol. 73, no. 2, p. 026605, 2006.

[47] F. B. Craig, F. Bohren, and D. Huffman, "Absorption and scattering of light by small particles," *Inc: John Wiley & Sons, New York, USA*, 1983.

[48] M. Born *et al.*, "E. wolf principles of optics," *Pergamon Press*, vol. 6, pp. 188–189, 1980.

[49] P. Hoekstra and A. Delaney, "Dielectric properties of soils at uhf and microwave frequencies," *Journal of geophysical research*, vol. 79, no. 11, pp. 1699–1708, 1974.

[50] G. C. Topp, J. Davis, and A. P. Annan, "Electromagnetic determination of soil water content: Measurements in coaxial transmission lines," *Water resources research*, vol. 16, no. 3, pp. 574–582, 1980.

[51] M. C. Dobson, F. T. Ulaby, M. T. Hallikainen, and M. A. El-Rayes, "Microwave dielectric behavior of wet soil-part ii: Dielectric mixing models," *IEEE Transactions on Geoscience and Remote Sensing*, no. 1, pp. 35–46, 1985.

[52] N. R. Peplinski, F. T. Ulaby, and M. C. Dobson, "Dielectric properties of soils in the 0.3-1.3-ghz range," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 3, pp. 803–807, 1995.

[53] V. L. Mironov, M. C. Dobson, V. H. Kaupp, S. A. Komarov, and V. N. Kleshchenko, "Generalized refractive mixing dielectric model for moist soils," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 4, pp. 773–785, 2004.

[54] G. Grazzini, M. Pieraccini, F. Parrini, A. Spinetti, G. Macaluso, D. Dei, and C. Atzeni, "An ultra-wideband high-dynamic range gpr for detecting buried people after collapse of buildings," in *Proceedings of the XIII Internarional Conference on Ground Penetrating Radar*. IEEE, 2010, pp. 1–6.

[55] A. A. Lestari, A. G. Yarovoy, and L. P. Ligthart, "Adaptive wire bow-tie antenna for gpr applications," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 5, pp. 1745–1754, 2005.

[56] A. Giannopoulos, "Modelling ground penetrating radar by GprMax," *Construction and building materials*, vol. 19, no. 10, pp. 755–762, 2005.

[57] A. Graps, "An introduction to wavelets," *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1995.

[58] O. Lopera, N. Milisavljevie, D. Daniels, and B. Macq, "Time-frequency domain signature analysis of gpr data for landmine identification," in *Advanced Ground Penetrating Radar, 2007 4th International Workshop on*. IEEE, 2007, pp. 159–162.

[59] G. D. Sower and S. P. Cave, "Detection and identification of mines from natural magnetic and electromagnetic resonances," in *Detection Technologies for Mines and Minelike Targets*, vol. 2496. International Society for Optics and Photonics, 1995, pp. 1015–1025.

[60] 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.

[61] S. Lambot, M. Antoine, M. Vanclooster, and E. C. Slob, "Effect of soil roughness on the inversion of off-ground monostatic gpr signal for noninvasive quantification of soil properties," *Water Resources Research*, vol. 42, no. 3, 2006.

[62] B. Karlsen, H. B. Sorensen, J. Larsen, and K. B. Jakobsen, "Independent component analysis for clutter reduction in ground penetrating radar data," in *Detection and Remediation Technologies for Mines and Minelike Targets VII*, vol. 4742. International Society for Optics and Photonics, 2002, pp. 378–390.

[63] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[64] C. Warren, A. Giannopoulos, and I. Giannakis, "An advanced gpr modelling framework: The next generation of gprmax," in *2015 8th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*. IEEE, 2015, pp. 1–4.

[65] F. Auger, P. Flandrin, P. Gonçalvès, and O. Lemoine, "Time-frequency toolbox," *CNRS France-Rice University*, vol. 46, 1996.

[66] Y. C. Eldar, *Sampling Theory: Beyond Bandlimited Systems*. Cambridge University Press, 2015.

[67] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[68] M. A. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE transactions on signal processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[69] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 31, no. 2, pp. 210–227, 2009.

[70] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Hyperspectral image classification using dictionary-based sparse representation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, 2011.

[71] K. Huang and S. Aviyente, “Sparse representation for signal classification,” in *Advances in neural information processing systems*, 2007, pp. 609–616.

[72] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit,” *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[73] Y. Naderahmadian, S. Beheshti, and M. A. Tinati, “Correlation based online dictionary learning algorithm,” *IEEE Trans. Signal Process*, vol. 64, no. 3, pp. 592–602, 2016.

[74] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[75] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[76] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, “Least angle regression,” *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[77] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A survey of sparse representation: algorithms and applications,” *IEEE access*, vol. 3, pp. 490–530, 2015.

[78] W. Cheney and D. Kincaid, “Linear algebra: Theory and applications,” *The Australian Mathematical Society*, vol. 110, 2009.

[79] G. Davis, S. Mallat, and M. Avellaneda, “Adaptive greedy approximations,” *Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[80] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[81] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[82] M. R. Osborne, B. Presnell, and B. A. Turlach, “A new approach to variable selection in least squares problems,” *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.

[83] I. Drori and D. L. Donoho, “Solution of l 1 minimization problems by lars/homotopy methods,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3. IEEE, 2006, pp. III–III.

[84] T. Hesterberg, N. H. Choi, L. Meier, C. Fraley *et al.*, “Least angle and $\ell 1$ penalized regression: A review,” *Statistics Surveys*, vol. 2, pp. 61–93, 2008.

[85] M. Elad, *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[86] S. Arora, R. Ge, and A. Moitra, “New algorithms for learning incoherent and overcomplete dictionaries,” in *Conference on Learning Theory*, 2014, pp. 779–806.

[87] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–22, 2006.

[88] T. H. Vu and V. Monga, "Fast low-rank shared dictionary learning for image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5160–5175, 2017.

[89] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[90] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2691–2698.

[91] K. R. Varshney, M. Çetin, J. W. Fisher, and A. S. Willsky, "Sparse representation in structured dictionaries with application to synthetic aperture radar," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3548–3561, 2008.

[92] Y. Suo, M. Dao, U. Srinivas, V. Monga, and T. D. Tran, "Structured dictionary learning for classification," *arXiv preprint arXiv:1406.1943*, 2014.

[93] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.

[94] L. Li, S. Li, and Y. Fu, "Learning low-rank and discriminative dictionary for image classification," *Image and Vision Computing*, vol. 32, no. 10, pp. 814–823, 2014.

[95] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3501–3508.

[96] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *European Conference on Computer Vision*, 2012, pp. 186–199.

[97] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 623–634, 2014.

[98] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 543–550.

[99] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[100] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[101] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193.

[102] A. Agarwal, A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon, "Learning sparsely used overcomplete dictionaries," in *Conference on Learning Theory*, 2014, pp. 123–137.

[103] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.

[104] F. Giovanneschi, K. V. Mishra, M. A. Gonzalez-Huici, Y. C. Eldar, and J. H. Ender, "Online dictionary learning aided target recognition in cognitive gpr," *arXiv preprint arXiv:1701.03607*, 2017.

[105] M. Cetin, I. Stojanovic, O. Onhon, K. Varshney, S. Samadi, W. C. Karl, and A. S. Willsky, "Sparsity-driven synthetic aperture radar imaging: Reconstruction, autofocusing, moving targets, and compressed sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 27–40, 2014.

[106] H. Hongxing, J. M. Bioucas-Dias, and V. Katkovnik, "Interferometric phase image estimation via sparse coding in the complex domain," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2587–2602, 2015.

[107] K. Aberman and Y. C. Eldar, "Sub-Nyquist SAR via Fourier domain range-Doppler processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6228–6244, 2017.

[108] A. C. Gurbuz, J. H. McClellan, and W. R. Scott, "Compressive sensing of underground structures using GPR," *Digit. Signal Process.*, vol. 22, no. 1, pp. 66–73, 2012.

[109] A. C. Gurbuz, O. Teke, and O. Arikan, "Sparse ground-penetrating radar imaging method for off-the-grid target problem," *J. Electron. Imaging*, vol. 22, no. 2, p. 021007, 2013.

[110] F. Soldovieri, R. Solimene, L. L. Monte, M. Bavusi, and A. Loperte, "Sparse reconstruction from gpr data with applications to rebar detection," *IEEE transactions on instrumentation and measurement*, vol. 60, no. 3, pp. 1070–1079, 2010.

[111] M. I. Skolnik, *Radar handbook*, 3rd ed. McGraw-Hill, 2008.

[112] K. V. Mishra, A. Kruger, and W. F. Krajewski, "Compressed sensing applied to weather radar," in *IEEE International Geoscience and Remote Sensing Symposium*, 2014, pp. 1832–1835.

[113] D. Cohen and Y. C. Eldar, "Reduced time-on-target in pulse Doppler radar: Slow time domain compressed sensing," in *IEEE Radar Conference*, 2016, pp. 1–4.

[114] J. Akhtar, B. Torvik, and K. E. Olsen, "Compressed sensing with interleaving slow-time pulses and hybrid sparse image reconstruction," in *IEEE Radar Conference*, 2017, pp. 0006–0010.

[115] I. M. Chakravarti, L. R. G., and J. Roy, *Handbook of methods of applied statistics: Volume I.* John Wiley and Sons, 2004.

[116] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *The Annals of Mathematical Statistics*, pp. 642–669, 1956.

[117] P. Massart *et al.*, "The tight constant in the dvoretzky-kiefer-wolfowitz inequality," *The Annals of Probability*, vol. 18, no. 3, pp. 1269–1283, 1990.

[118] 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*, 2017, pp. 508–512.

[119] L. E. Besaw and P. J. Stimac, "Deep convolutional neural networks for classifying GPR B-scans," *Proceedings of SPIE*, vol. 9454, p. 945413, 2015.

[120] J. N. Wilson, P. Gader, W.-H. Lee, H. Frigui, and K. Ho, "A large-scale systematic evaluation of algorithms using ground-penetrating radar for landmine detection and discrimination," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2560–2572, 2007.

[121] E. D. Sontag, "VC dimension of neural networks," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 69–96, 1998.

[122] J. Leckebusch, "Comparison of a stepped-frequency continuous wave and a pulsed gpr system," *Archaeological Prospection*, vol. 18, no. 1, pp. 15–25, 2011.

[123] M. G. M. Hussain, "Principles of high-resolution radar based on nonsinusoidal waves - Part III: Radar-target reflectivity model," *IEEE Transactions on Electromagnetic Compatibility*, vol. 32, no. 2, pp. 144–152, 1990.

[124] D. J. Daniels, *Ground penetrating radar*. IET, 2004.

[125] D. Pasculli and G. Manacorda, "Real-time, pseudo real-time and stroboscopic sampling in time-domain gprs," in *Advanced Ground Penetrating Radar (IWAGPR), 2015 8th International Workshop on*. IEEE, 2015, pp. 1–4.

[126] A. Bystrov and M. Gashinova, "Analysis of stroboscopic signal sampling for radar target detectors and range finders," *IET Radar, Sonar & Navigation*, vol. 7, no. 4, pp. 451–458, 2013.

[127] I. Chant, D. Lee, and D. Ireland, "Dsto landmine detection test targets," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS . . . , Tech. Rep., 2005.

[128] J. Chen, L. Jiao, W. Ma, and H. Liu, "Unsupervised high-level feature extraction of SAR imagery with structured sparsity priors and incremental dictionary learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 10, pp. 1467–1471, 2016.

[129] V. Glivenko, "Sulla determinazione empirica della legge di probabilità," *Giornale Dell'Istituto Italiano Degli Attuari*, no. 4, p. 92–99, 1933, in Italian.

[130] F. P. Cantelli, "Sulla determinazione empirica della legge di probabilità," *Giornale Dell'Istituto Italiano Degli Attuari*, no. 4, p. 221–424, 1933, in Italian.

[131] G. Lillacci and M. Khammash, “Model selection in stochastic chemical reaction networks using flow cytometry data,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 1680–1685.

[132] T. Vu, L. Nguyen, T. Guo, and V. Monga, “Deep network for simultaneous decomposition and classification in uwb-sar imagery,” *arXiv preprint arXiv:1801.05458*, 2018.

[133] R. Girshick, “Fast R-CNN,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[134] I. Giannakis, A. Giannopoulos, and C. Warren, “A realistic fdtd numerical modeling framework of ground penetrating radar for landmine detection,” *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 9, no. 1, pp. 37–51, 2016.

[135] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[136] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[137] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[138] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[139] M. Shaw, S. Millard, T. Molyneaux, M. Taylor, and J. Bungey, “Location of steel reinforcement in concrete using ground penetrating radar and neural networks,” *Ndt & E International*, vol. 38, no. 3, pp. 203–212, 2005.

[140] P. Gamba and S. Lossani, “Neural detection of pipe signatures in ground penetrating radar images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 2, pp. 790–797, 2000.

[141] T. Saarenketo and T. Scullion, “Road evaluation with ground penetrating radar,” *Journal of applied geophysics*, vol. 43, no. 2-4, pp. 119–138, 2000.

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

[143] W. Al-Nuaimy, Y. Huang, M. Nakhkash, M. Fang, V. Nguyen, and A. Eriksen, “Automatic detection of buried utilities and solid objects with gpr using neural networks and pattern recognition,” *Journal of applied Geophysics*, vol. 43, no. 2-4, pp. 157–165, 2000.

[144] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[145] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, “A practical guide to support vector classification,” 2003.

[146] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intelligent Syst. Tech.*, vol. 2, no. 3, p. 27, 2011.

Appendices

Appendix A

Convolutional Neural Networks

Neural networks, also known as Artificial Neural Networks (ANN), are computational models which have been studied and developed since 1943 thanks to the early works of McCulloch, Warren and Pitts [136]. ANN are widely used for speech recognition, computer vision, text processing and classification tasks in general; they belong to the branch of Machine Learning research called *Deep Learning* which is experiencing a lot of contributions in the last years [137, 138]. The computational model of a neural network is inspired by how the human brain process information; the basic computation unit of an ANN is called neuron (or node). A neuron receives one or more inputs ($x_1 ... x_n$) and process one output (y) based on an activation function f. Since real world data is generally not linear, and we want the network to be able to learn this non-linearity behaviour, f is a non-linear function. The inputs of f are weighted according to the relevance that we want to give to each one of them, plus there can be a bias weight b associate to an additional input. As an example, for a node with two inputs (x_1 and x_2) with their respective weights (w_1 and w_2) and a bias factor b we get the following activation function: $y = f(w_1 * x_1 + w_2 * x_2 + 1 * b)$. Popular activation functions being used in pratice are the sigmoid ($\sigma(x) = \frac{1}{1+e^{-x}}$), the hyperbolic tangent ($\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$) and the Rectified Linear Unit, ReLu ($f(x) = max(0, x)$). The simplest form of neural networks has a feed-forward structure in the sense that the information moves only from the input to the output of the network (i.e. there are no cycles or loops). In its simplest form, a feed-forward neural network has an input layer and an output layer (Single layer perceptron), if it contains one or more layers of neurons in between the input and the output layers (called hidden layers) it's called Multi Layer Perceptron (MLP). Every layer of the network contains a pre-defined number of neurons; the neurons in the input layer just pass the data from the outside world to the network. In the hidden layer, the neurons compute the input data and move it forward to the output layer or progress to more hidden layers. The neurons in the output layer process the ouputs from the hidden layers to generate the output of the neural network. The output nodes usually employ an activation function called *softmax* which ensures that the output probabilities of the ANN sums up to 1. Figure A.1 shows a basic structure of a MLP with three input node (and one bias node), one hidden layer (with one bias node), and two output nodes. The training of a neural network is the process where we learn the weights for every node in every layer according to a training data of which we know already the output (i.e. the training set is labeled, this procedure is called *supervised learning*). Once the learning phase is over, the neural network is able to

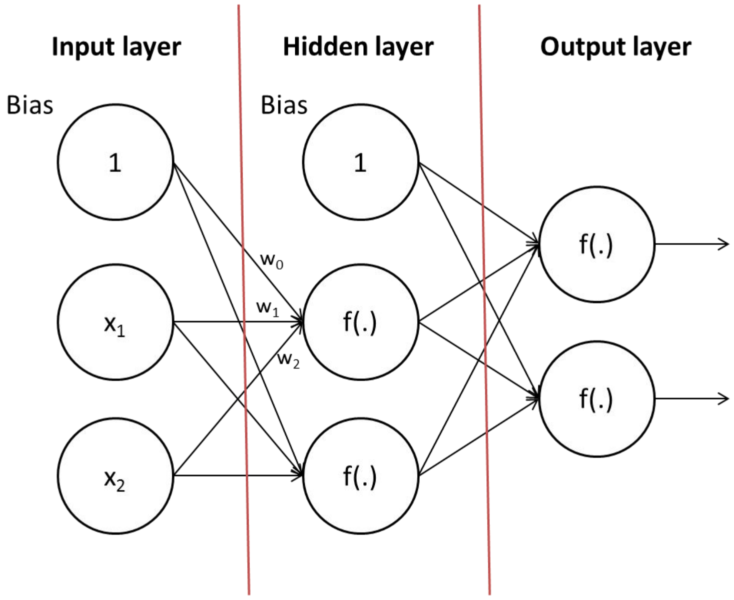


Figure A.1 – A example of Multi Layer Perceptron with one hidden layer

predict the output for new data (test data). A MLP can be trained using a *backpropagation* algorithm. Initially the weights are assigned randomly, then the ANN is activated for each input in the training set. The error between the known output and the correct one is propagated back to the previous layers and the weights are adjusted accordingly for every neuron, the most popular strategy to adjust the weights is the *gradient descent* [137].

Convolutional Neural Networks (CNN) are known to efficiently exploit structural or locational information in the data and yield comparable learning potential using far fewer parameters with respect to ANN [132, 133]. In this subsection we briefly describe their basic structure and operation.

In ANN, the activations are generated by direct matrix multiplication; every neuron process all the input from the previous layer, using a different weight for each one of them; the layers in an ANN are *fully connected*. For high-dimensional datasets (such as time series or images) this can result in an huge amount of weights, especially when the number of neurons in the network increases. This will also slow down the computations and lead to a higher chanche of *overfitting* [135] CNN employs *convolutional layers*; these layers use convolution (instead of matrix multiplication) with *filters* (also called kernels) of different dimension to extract different features of the data at every layer. It is very common to use two-dimensional or three dimensional images (RGB) as an input of the neural network for image classification purposes. However, since the data that we will deal with are one-dimensional time series, we will refer to such representations for the following examples. The convolution operation between a one dimensional time series $x(t)$ and a filter $f(t)$ filter can be written like this:

$$y(t) = x(t) * f(t) = \sum_{a=-\infty}^{+\infty} x(a) f(t-a), \tag{A.1}$$

where $y(t)$ is the output of the convolution operation, in CNN, its samples are called *activations*. One can apply different filters to the input data and obtain more outputs, i.e. *channels*; the dimension of the filters and their number are one of the *hyperparameters* of a CNN layer. Two other important hyperparameters of a convolutional layer are the *stride* and the *padding*. The stride indicates the amount of shift (in terms of data samples) between two different convolutions, while the padding regulates what the amount of zeros to be added at the borders of the data before computing convolution. Typical padding parameters are *same* (where we add zeros in order to have the number of activations equal to the input data) and *valid* (where we add no zeros at the border and therefore the output data ,i.e. the activations, will have a smaller dimension). Another building block of a convolutional layer is the *pooling* which in many texts it is considered an additional layer, here we will treat it as a block inside the actual convolutional layer. Pooling needs no hyperparameters apart from the filter dimension and the stride, it can be seen as an additional filter which, instead of performing the convolution operation with a filter, it just take the maximum (maxpool) or the average of the considered activations from the previous layer. A convolutional neural network can still include fully connected layers in its architecture and the softmax function for the output layer is still used. The non-linear activation function (such as the ReLu) and the bias are still employed in CNN.

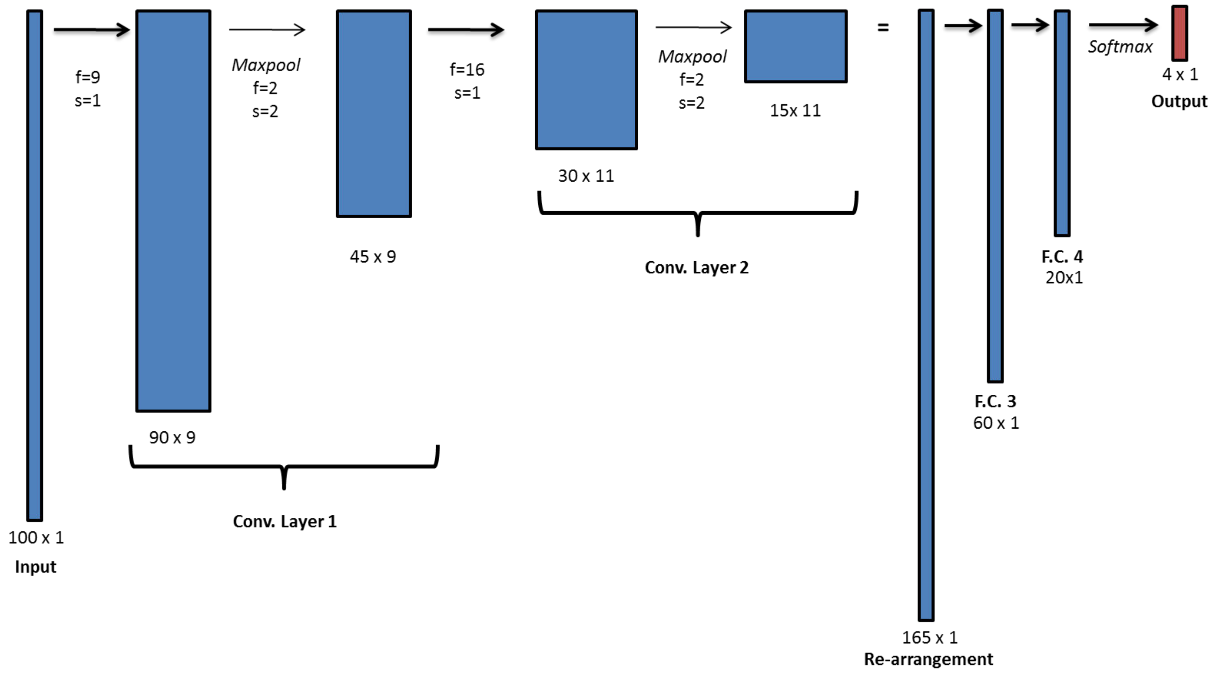


Figure A.2 – An example of a typical CNN architecture

Figure A.2 shows an example of a typical architecture of a CNN with two convolutional layers, two fully connected layers and a softmax function for the output. We assumed a valid padding in the two convolutional layers and a softmax function for obtain a vector of four outputs. The relation that gives the dimension of the next layer, according to the previous input size is given by:

$$n_{out} = \frac{n_{in} + 2p - f}{s} + 1, \tag{A.2}$$

where n_{in} and n_{out} are the input and output size respectively, p is the padding, f is the filter dimension and s is the stride.

Neural networks have been previously employed with GPR data for classifying different types targets/structures: steel reinforcements [139], underground services [140], road evaluation [141], buried landmines [120, 142] and more [143].

Appendix B

Support Vector Machines

Support Vector Machines are an useful tool for data classification [144–146]. Like neural networks and many other classification approaches, SVM relies on the presence of a training set of known observations and a test set of unknown ones. SVM work in two steps, first they learn a *functional* from the labeled training set, then this functional is used to assign every observation to a particular class. The functional works on a *feature space* ($\mathcal{F}$) of higher dimension respect to the original space of the observations; here the data can be separated more easily. The second step is to use the obtained functional to classify unknown data. It is obvious that, for satisfactory classification results, the training set has to be representative of the test set that we want to classify. Let $\mathbf{X} \in \mathbb{R}^{mxl}$ with $x_i \in \mathbb{R}^m$ and $i = 1...l$ be a pre-defined collection of l labeled observations, namely a *training set*, SVM searches for a functional which will assign a class to any given observation x_i. Assuming a 2-class classifier we have then:

$$f : \mathbb{R}^m \to \mathbb{R} \begin{cases} f(\mathbf{x}_i) \geq 0, & \text{assign observation to class } +1 \\ f(\mathbf{x}_i) \leq 0, & \text{assign observation to class } -1 \end{cases} \tag{B.1}$$

SVM moves the data into the aforementioned feature space ($\mathcal{F}$), by the means of a function Φ. In this high-dimensional space, a simple *hyperplane* will suffice for the separation between different classes. An hyperplane can be described as the set of points of $\mathbf{x}$ satisfying $\mathbf{w} \cdot \mathbf{x} - b = 0$, with $\mathbf{w}$ being a vector normal to the hyperplane and b an offset parameter.

$$\Phi : \mathbb{R}^n \to \mathcal{F}, \mathbf{x} \to \Phi(\mathbf{x}) \tag{B.2}$$

It is possible to map observations from a general set into a feature space $\mathcal{F}$ without having to compute the mapping explicitly if only dot products are used between vectors in the feature space. These high-dimensional dot products can be computed within the original space, by means of a kernel function; K is a Kernel if exists a mapping Φ such that:

$$K(x_i, x_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \tag{B.3}$$

Where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two observation vectors in the original set. One popular choice for kernel functions is the Radial Basis Function (RBF).

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)} \tag{B.4}$$

Being $f(\mathbf{x}) = \mathbf{wx} + \mathbf{b}$ the simple hyperplane which separates 2 different classes in the feature space (with the associated parameters $\mathbf{w}$ and $\mathbf{b}$) and being $\mathbf{y} \in \mathbb{R}^l$ the indicator vector such that $\mathbf{y}_i \in -1, 1$ SVM solves the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{w},\mathbf{b},\chi}{\text{minimize}} \;\; \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \xi_i \\ &\text{subject to} \;\; \mathbf{y}_i(\mathbf{w}^T\Phi(\mathbf{x}_i) + \mathbf{b}) \geq 1 - \xi_i, \\ &\text{with} \;\; \xi_i \geq 0 \end{aligned} \tag{B.5}$$

Where $C \geq 0$ is the penalty parameter of the error term. The solution of this problem can be tuned according to the selection of the parameters C and, assuming we are using a RBF kernel, the parameter γ. The correct values for C and γ are usually found by a cross-validation process. Cross validation is about splitting the training set (for wich we know exactly the labels of each class) into two (or more) subsets. One subset will be used to train the SVM, i.e. to obtain the functional f for each combination pair of C and γ among an empirically selected range of values. The other subset(s) will be used as validation to test the classification accuracy with the obtained functional. By refining the range of values for C and γ one can find the combination which gives the best classification accuracy for one particular training set. To further improve the selection of the parameters, one can use a n-fold cross validation where the process is averaged over n different combinations of training and validation vectors.

Support vector machines have been succesfully applied for GPR target classification [40, 41, 104] and they will also be a key ingredient of our adaptive sparse-representation-based approach.

Acronyms

APM - Antipersonnel landmines
ADMM - Alternating Direction Method of Multipliers
dB - Decibel
BP - Basis Pursuit
BPDN - Basis Pursuit De-Noise
CNN - Convolutional Neural Network
CBWLSU - Correlation Based Weighted Least Square Update
DL - Dictionary Learning
CDF - Cumulative Distribution Function
CV - Coefficient of Variation
DOMINODL - Drop-Off Mini-Batch Online Dictionary Learning
DKW - Dvoretzky-Kiefer-Wolfowitz
ECDF - Empirical Cumulative Distribution Function
ERA - Electrical Research Association
EPDF - Empirical Probability Distribution Function
ERW - Explosives Remnants of War
FDTD - Finite Diference Time Domain
FFT - Fast Fourier Transform
GPR - Ground Penetrating Radar
HMM - Hidden Markov Model
IED - Improvised Explosives Devices
RCS - Radar Cross Section
RBF - Radial Basis Function
KS - Kolmogorov Smirnoff
LARS - Least Angle Regression
LASSO - Least Absolute Shrinkage and Selection Operator
LIAG - Liebniz Institute for Applied Geophysics
LS - Least Square
LRSDL - Low Rank Shared Dictionary Learning
MOD - Method of Optimal Directions
NRMSE - Normalized Root Mean Square error

NN - Neural Networks

ODL - Online Dictionary Learning

OMP - Orthogonal Matching Pursuit

PDF - Probability Distribution Function

PLS - Penalized Least Square

RIP - Restricted Isometry Property

SNR - Signal to Noise Ratio

SR - Sparse Representation

SRC - Sparse Representation Based Classification

SVM - Support Vector Machine

SVD - Singular Values Decomposition

UWB - Ultra Wideband

UXO - Undexploded Ordnance

WLS - Weighted Least Squares

List of Figures

List of Tables

Declaration

I hereby declare that the present work was done by myself and no other than the cited sources were employed.

Ich erkläre mich hiermit, die vorliegende Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertig zu haben.

Siegen, December 2018.

Fabio Giovanneschi