

# Boosted Classifiers for Antitank Mine Detection in C-Scans from Ground-Penetrating Radar

Przemysław Klęsk, Mariusz Kapruziak and Bogdan Olech

**Abstract** We investigate the problem of automatic antitank mine detection. Subject to detection are 3D images, so-called C-scans, generated by a GPR (Ground-Penetrating Radar) system of our construction. In the paper we focus on *boosting* as a machine learning approach well suited for large-scale data such as GPR data. We compare five variants of weak learners with real-valued responses trained by the same boosting scheme. Three of the variants are single-feature-based learners that differ in the way they approximate class conditional distributions. The two remaining variants are shallow decision trees, respectively, with four and eight terminal nodes, introducing joint-feature conditionals.

**Keywords** Boosting · Weak learners · Ground Penetrating Radar · C-scans · Landmine detection

---

This project has been partially financed by the Polish Ministry of Science and Higher Education; agreement no. 0091/R/TOO/2010/12 for R&D project no. 0 R00 0091 12, dated on 30.11.2010, carried out by a consortium of Military Institute of Armament Technology in Zielonka, and Autocomp Management in Szczecin, Poland.

---

P. Klęsk (✉) · M. Kapruziak · B. Olech  
Faculty of Computer Science, West Pomeranian University of Technology,  
ul. Żołnierska 49, 71-210 Szczecin, Poland  
e-mail: pklesk@wi.zut.edu.pl

M. Kapruziak  
e-mail: mkapruziak@wi.zut.edu.pl

B. Olech  
e-mail: bolech@wi.zut.edu.pl

M. Kapruziak · B. Olech  
Autocomp Management, ul. Władysława IV 1, 70-651 Szczecin, Poland

# 1 Introduction

Postconflict areas are typically contaminated by explosive remnants such as unexploded ordnance (UXO) and landmines, both antipersonnel (AP) and antitank (AT). United Nations report that about 80 countries are burdened with this problem (see <http://www.un.org/en/globalissues/demining/>). The process of bringing such areas back to normal use is slow and expensive [18].

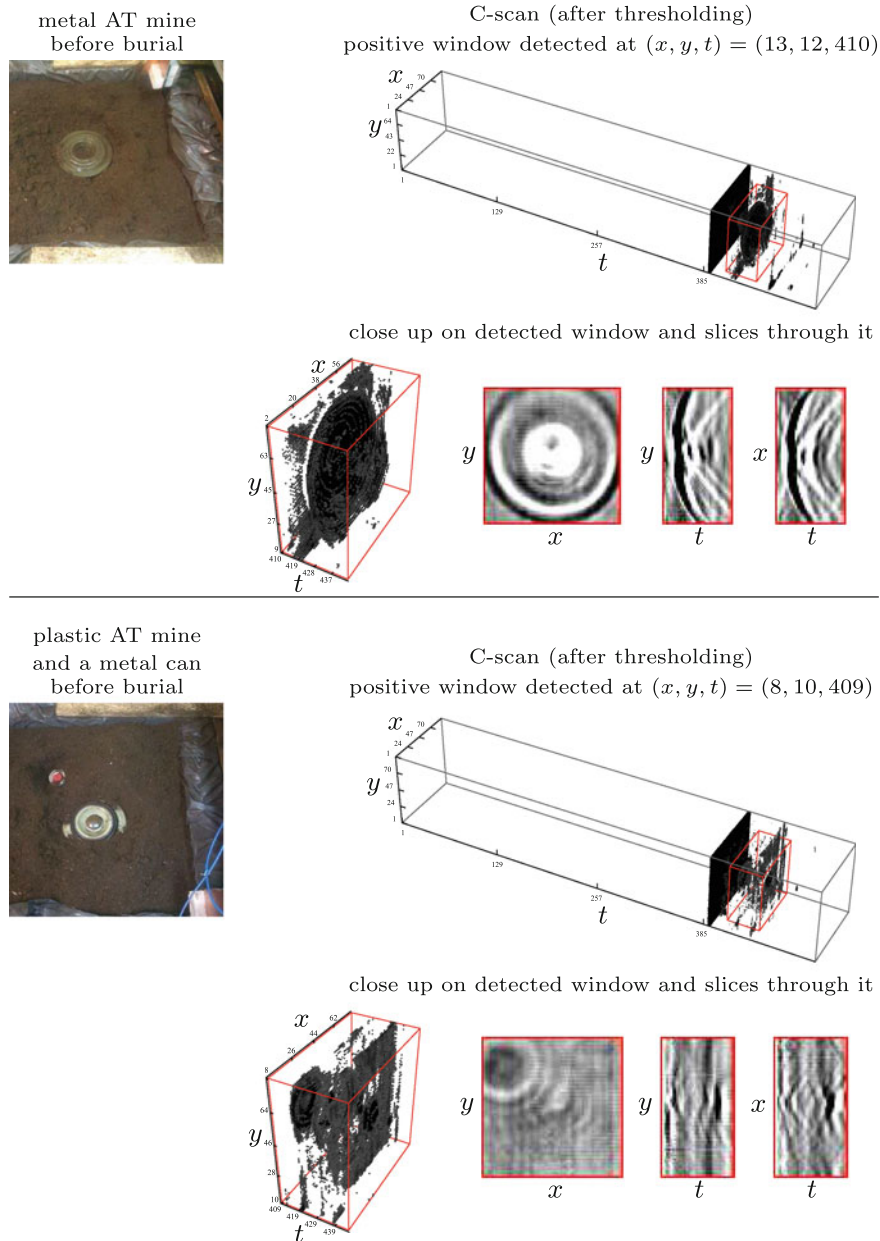
As regards the demining technology, up to recently the electromagnetic induction (EMI) metal detector, often manual, has been the most frequent tool. Unfortunately, it is of little use for mines with low metal content (plastic, wooden, or glass mines), and moreover has a high false alarm rate (FAR)—it is estimated that about  $10^3$  false positives go on average with each correctly detected mine. The last decade brought a progress in the technology of Ground-Penetrating Radar, a method working with frequencies up to 20 GHz which potentially enables for high-resolution imaging and “seeing” nonmetallic objects. GPR-based mine detection is still far from ideal, as GPR is susceptible to all kinds of inhomogeneities and clutter in the ground (stones, roots, bricks, water pockets, etc.) and hence produces some false alarms as well, but results reported in the literature are much superior than the ones for EMI detectors [7, 18].

## 1.1 GPR Images and Mine Detection Approaches

A C-scan is a 3D image built up as a collection (taken over some area) of A-scans, i.e., one-dimensional signals generated by GPR over an axis directed into the ground (time axis  $t$ ). A boundary of any buried object with dielectric properties different from the medium (soil) leaves a mark in an A-scan. As the GPR system moves across track ( $x$ -axis) or along track ( $y$ -axis), while taking successive A-scans, the marks generate a *hyperbola* trace in the image. Hence, an object seen in a C-scan resembles a combination of *hyperboloids*. The detection task is therefore about distinguishing mine-related hyperboloids from others. In the Fig. 1 we show detection examples in C-scans collected by our GPR vehicle.

Mine detection approaches met in the literature are often based on machine learning, and typically focus on two aspects: features extraction and learning algorithms. In majority of cases, extraction of features is done from preselected 2D projections (B-scans or time slices), and rather seldom directly from C-scans. Some authors try to reconstruct the physical features from an image (e.g., depth of burial, radius, height, dielectric permittivity of the ground) [19], however, the majority of works focuses on simpler and purely image-oriented features. By that meant are, e.g., histograms, statistical moments, principal components (PCA), structure-based features, edge, or shape-related descriptors [4, 9, 16].

In mentioned works, commonly a fairly small number of features is used to train classifiers—from several up to several tens. A notable exception can be met in [16]



**Fig. 1** Two examples of successful detections of metal AT mine (*upper part*) and plastic AT mine (*bottom part*). Input C-scans taken over approx. 1 m<sup>2</sup> area

due to Torrione and Collins, where authors adopted the *Texture Feature Coding Method* [6]. For given pixel of a C-scan they consider 13 difference—vectors crossing it within the  $3 \times 3$  neighborhood. The vectors are then quantized into: {drop, no-change, growth}, and a codebook of textures is formed. By repeating such operations on many image fragments, potentially thousands of features can be generated. The authors report to limit themselves to 560 at maximum.

As regards the learning algorithms, quite many of them have been tried out in mine detection applications, e.g.: Naive Bayes and LVQ [2], neural networks [4], HMMs [4, 11], SVMs [5], RVMs [16]. It is difficult to compare fairly the results due to different experimental settings (types of mines used, soil types, amount of clutter, weather, etc.), nevertheless, to give a general view on recent achievements (e.g., [4, 5, 11, 14, 16])—commonly reported are detection rates (sensitivities) ranging approximately from 93 to 98 % with FARs typically not greater than 0.05 FA/m<sup>2</sup>, and reported AUC<sup>1</sup> measures ranging within 95–99.5 %.

## 1.2 Boosting and Our Motivation

The idea of boosting, proposed first by Schapire [13], is considered to be a crucial discovery of recent years in machine learning. Boosting is a meta-method. It works by sequentially applying some simple learning algorithm to reweighted versions of the training data, thus forming an ensemble of partial classifiers. The final response is taken as a majority or weighted vote. Known properties of boosting are its: (1) suitability for large-scale data, (2) ability to automatically select relevant features, and (3) robustness to overfitting (good generalization)—practical applications indicate that as new classifiers are added the test error decreases and then levels off, instead of ultimately increasing. Friedman, Hastie, and Tibshirani [3] have demonstrated that boosting can be viewed and understood as an *additive model* for *logistic regression*, explaining some of the “mystery” why boosting works well (see notes in the Appendix A).

In the context of mine detection, boosting might be of importance if one decides for a “mass attack” approach consisting in generating huge sets of simple features (e.g., of size  $\sim 10^4$ ) from GPR images and letting the algorithm discover a smaller relevant subset (e.g., of size  $\sim 10^2$ ). Note that such an approach worked very well in face detection applications, where ideas due to Viola and Jones [17] based on multiple Haar-like features are becoming a standard. Apart from our previous work [8] we have managed to come across only two papers [14, 15] on mine detection with the traditional AdaBoost being the learning algorithm.<sup>2</sup>

In this paper we apply the mentioned approach (at the training stage we generate over 15 thousand features—being 3D Fourier moments) and our main focus is to compare the performance of several variants of weak learners.

<sup>1</sup> Area under Receiver Operating Characteristic curve.

<sup>2</sup> Yet, the authors in [14, 15] consider quite small sets of about 20 features.

## 2 Boosting and Weak Learners to Be Compared

### 2.1 Notation

Let  $\{(\mathbf{x}_i, y_i)\}_{i=1,\dots,m}$  denote the set of training examples described by vectors of features  $\mathbf{x}_i = (x_{i1}, \dots, x_{id}) \in \mathbb{R}^d$  and class labels  $y_i \in \{-1, 1\}$ . In our context,  $\mathbf{x}_i$  vectors represent features extracted from a 3D window being a part of a C-scan, and the positive class ( $y_i = 1$ ) informs about a mine present in the window.

### 2.2 Boosting Scheme for Real-Valued Weak Learners

Beneath, we present a general boosting scheme in which weak learners are assumed to have real-valued responses (*real boost*). The number of rounds is denoted by  $K$ , weak classifiers by  $f_k$ , and the ensemble by  $F$ .

1. Start with uniform weights on data examples  $w_i := 1/m, i = 1, \dots, m$ .
2. For  $k = 1, \dots, K$  repeat:
  - 2.1. Train a new weak classifier  $f_k$  using weights  $w_i$  on the training data with the goal to minimize the *exponential criterion*  $\sum_{i=1}^m w_i e^{-y_i f_k(\mathbf{x}_i)}$ , or equivalently so that  $f_k(\mathbf{x})$  is an approximation of half the logit transform:

$$f_k(\mathbf{x}) := \frac{1}{2} \log \left( \hat{P}_w(y = 1|\mathbf{x}) / \hat{P}_w(y = -1|\mathbf{x}) \right). \quad (1)$$

- 2.2. Update the weights:

$$Z_k := \sum_{i=1}^m w_i e^{-y_i f_k(\mathbf{x}_i)}; \quad w_i := w_i e^{-y_i f_k(\mathbf{x}_i)} / Z_k, \quad i = 1, \dots, m. \quad (2)$$

3. The final (ensemble) classifier is  $F(\mathbf{x}) := \sum_{k=1}^K f_k(\mathbf{x})$  with the decision returned according to  $\text{sgn } F(\mathbf{x})$ .

### 2.3 Compared Variants of Weak Learners

#### 2.3.1 NormalRealBoost (NRB)

This learner iterates over all features  $x_j$  and approximates the conditional densities  $p(x_j | y = \pm 1)$  by *normal* densities  $\hat{p}_w(x_j | y = \pm 1) = 1 / \sqrt{2\pi \sigma_{j\pm}^2} e^{-(x_j - \mu_{j\pm})^2 / (2\sigma_{j\pm}^2)}$  with means and variances calculated as:

$$\mu_{j-} = \sum_{\substack{i=1 \\ y_i=-1}}^m w_i x_{ij} / \sum_{\substack{i=1 \\ y_i=-1}}^m w_i, \quad \mu_{j+} = \sum_{\substack{i=1 \\ y_i=1}}^m w_i x_{ij} / \sum_{\substack{i=1 \\ y_i=1}}^m w_i, \quad (3)$$

$$\sigma_{j-}^2 = \sum_{\substack{i=1 \\ y_i=-1}}^m w_i x_{ij}^2 / \sum_{\substack{i=1 \\ y_i=-1}}^m w_i - \mu_{j-}^2, \quad \sigma_{j+}^2 = \sum_{\substack{i=1 \\ y_i=1}}^m w_i x_{ij}^2 / \sum_{\substack{i=1 \\ y_i=1}}^m w_i - \mu_{j+}^2. \quad (4)$$

By means of Bayes' theorem the response of this weak learner is calculated as:

$$f_k(\mathbf{x}) = \frac{1}{2} \log \frac{\widehat{P}_w(x_{j^*}|y=1)\widehat{P}_w(y=1)}{\widehat{P}_w(x_{j^*}|y=-1)\widehat{P}_w(y=-1)} \quad (5)$$

$$= \frac{1}{2} \left( \frac{(x_{j^*} - \mu_{j^*-})^2}{2\sigma_{j^*-}^2} - \frac{(x_{j^*} - \mu_{j^*+})^2}{2\sigma_{j^*+}^2} + \log \frac{\sigma_{j^*-}}{\sigma_{j^*+}} + \log \frac{\widehat{P}_w(y=1)}{\widehat{P}_w(y=-1)} \right), \quad (6)$$

where  $\widehat{P}_w(y = \pm 1) = \sum_{\{i: y_i = \pm 1\}} w_i$  are current class probability estimates and  $j^*$  indicates the feature for which the exponential criterion is the smallest.

### 2.3.2 BinningRealBoost (BRB)

The idea behind this learner is similar to NRB, with the difference that conditional densities for each feature are approximated by *piecewise constant* functions instead of normals. Such functions are typically implemented by a binning mechanism, see e.g., [8, 12].

Suppose that  $[a_1, a_2]$  represents the range of some features. Let  $B$  denote the wanted number of bins. For convenience we use bins of equal widths. The bin index  $\beta(x) \in \{1, \dots, B\}$  that an argument  $x$  belongs to is:  $\beta(x) = \lceil B(x - a_1) / (a_2 - a_1) \rceil$  for  $a_1 < x \leq a_2$ ; with border cases:  $\beta(x) = 1$  for  $x \leq a_1$  and  $\beta(x) = B$  for  $a_2 < x$ . Let  $\widehat{P}_w(y = -1, j \text{ in } b) = \sum_{\{i: y_i = -1, \beta(x_{ij}) = b\}} w_i$  denote the estimated probability that an example is negative and its  $j$ th feature belongs to the bin  $b$ . Then, the response of this weak classifier using the best  $j^*$ th feature is:

$$f_k(\mathbf{x}) = \frac{1}{2} \log \left( \widehat{P}_w(y = 1, j^* \text{ in } \beta(x_{j^*})) / \widehat{P}_w(y = -1, j^* \text{ in } \beta(x_{j^*})) \right). \quad (7)$$

In our experiments we shall test two variants named BRB4 and BRB8 where the number of bins  $B$  is four and eight, respectively.

### 2.3.3 DecisionTreesRealBoost (DTRB)

This learner is based on the well-known CART algorithm [1]. Practical and theoretical evidences show that an ensemble of many shallow decision trees generalizes better than a deep single tree [3, 10].

In experiments we test two variants named DTRB4 and DTRB8 representing trees with four and eight terminal nodes, respectively. The generation of each tree has been carried out in a traditional fashion using Gini index as the impurity criterion. Not to repeat the whole algorithm, hereby we describe only the splits evaluation. Suppose we are at a certain stage of the recursion and the set of indices  $\{i\}$  is restricted only to data examples falling into the given tree node. Quantities of interest for a candidate split on the  $j$ th feature and a value  $v$  are:

$$\begin{aligned}
 W(L) &= \sum_{\{i : x_{ij} < v\}} w_i, & W(R) &= \sum_{\{i : x_{ij} \geq v\}} w_i, \\
 W(y=-1, L) &= \sum_{\{i : x_{ij} < v, y_i = -1\}} w_i, & W(y=1, L) &= \sum_{\{i : x_{ij} < v, y_i = 1\}} w_i, \\
 W(y=-1, R) &= \sum_{\{i : x_{ij} \geq v, y_i = -1\}} w_i, & W(y=1, R) &= \sum_{\{i : x_{ij} \geq v, y_i = 1\}} w_i, \quad (8)
 \end{aligned}$$

where  $L$  and  $R$  denote, respectively, the left and the right part after the split is made, and the suitable probability estimates can be calculated as follows:

$$\begin{aligned}
 \hat{P}_w(L) &= W(L) / (W(L) + W(R)), & \hat{P}_w(R) &= W(R) / (W(L) + W(R)), \\
 \hat{P}_w(y=-1|L) &= W(y=-1, L) / W(L), & \hat{P}_w(y=1|L) &= W(y=1, L) / W(L), \\
 \hat{P}_w(y=-1|R) &= W(y=-1, R) / W(R), & \hat{P}_w(y=1|R) &= W(y=1, R) / W(R).
 \end{aligned} \quad (9)$$

Then, the expected Gini impurity of the split is:

$$\hat{P}_{w(L)} (1 - \hat{P}_w^2(y=-1|L) - \hat{P}_w^2(y=1|L)) + \hat{P}_{w(R)} (1 - \hat{P}_w^2(y=-1|R) - \hat{P}_w^2(y=1|R)). \quad (10)$$

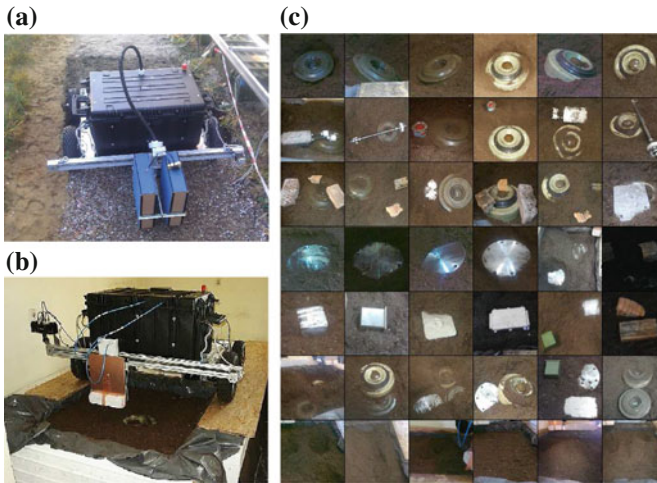
Each terminal node of a tree responds with a real value in accordance with half the logit function (1), i.e.,  $\frac{1}{2} \log(\sum_{\{i : y_i = 1\}} w_i / \sum_{\{i : y_i = -1\}} w_i)$ .

### 3 Experiments

#### 3.1 Learning Material

Our GPR-equipped vehicle (see the Fig. 2) was initially tested in outdoor conditions (over peat, garden, gravel, sand soils) and proved to produce sufficiently clear images. For convenience, the main learning material was collected in indoor conditions on a laboratory test stand with garden soil. Scanned scenes involved two types of AT mines (metal and plastic) and a variety of disruptive objects (cans, boxes, bricks, discs, shafts, cables). We experimented with different burial depths, lean angles, surface variations, “mine imitations” set up by suitable arrangements of nonmine objects; some scene examples are depicted in the Fig. 2c.

The final material consisted of 210 high-resolution C-scans (1 cm both across and along track, 512 time samples), each taken from an area of  $\approx 1 \text{ m}^2$ . The material included: 70 scans with the metal AT mine (and possibly other objects), 70 scans with the plastic AT mine (and possibly other objects), 70 scans with nonmine objects only. The scans were then divided into train (80 %) and test (20 %) parts, with the test part consisting of: 14 metal mine scans, 14 plastic mine scans, and 14 scans with nonmines. Finally, all scans were traversed by a sampling 3D window of widths  $(w_x, w_y, w_t) = (63, 63, 35)$  and  $(w_x, w_y, w_t) = (72, 72, 40)$ —two passes over each C-scan, and a data set of positive and negative window examples was created for supervised learning. We have memorized all positive windows, whereas negative windows were undersampled. This resulted in a  $\approx 4 \text{ GB}$  train set with about  $2.1 \times 10^3$  positives and  $6.8 \times 10^4$  negatives; and a  $\approx 1 \text{ GB}$  test set with  $5.5 \times 10^2$  positives and



**Fig. 2** GPR vehicle of our construction shown outdoor (a) and on a laboratory test stand (b). Examples of scanned scenes and variations (c)



$1.7 \times 10^4$  negatives. Each window example was described by 15,625 shape-related features.

### 3.2 Features Extraction

To extract features we applied *piecewise 3D Fourier approximations* (of low orders) with respect to image windows. Each window was divided into a regular grid of 125 cuboids ( $5 \times 5 \times 5$ ), and for each cuboid, spanning in general from  $(x_1, y_1, t_1)$  to  $(x_2, y_2, t_2)$ , Fourier coefficients (3D moments) were calculated as:

$$c_{k_x, k_y, k_t} = \frac{\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} \sum_{t=t_1}^{t_2} i(x, y, t) e^{-2\pi i \left( k_x \frac{x-x_1}{x_2-x_1+1} + k_y \frac{y-y_1}{y_2-y_1+1} + k_t \frac{t-t_1}{t_2-t_1+1} \right)}}{(x_2 - x_1 + 1)(y_2 - y_1 + 1)(t_2 - t_1 + 1)}, \quad (11)$$

where:  $-n \leq k_x, k_y, k_t \leq n$  indicate harmonic orders (variable-wise) of the coefficient with  $n$  being the maximum,  $i(x, y, t)$  is the pixel intensity at a particular point of the C-scan, and  $i = \sqrt{-1}$  is the imaginary unit. Then, real and imaginary parts of the coefficients were taken as the features.<sup>3</sup>

Due to the well-known symmetry property:  $\text{Re } c_{k_x, k_y, k_t} = \text{Re } c_{-k_x, -k_y, -k_t}$ ,  $\text{Im } c_{k_x, k_y, k_t} = -\text{Im } c_{-k_x, -k_y, -k_t}$ , and the fact that  $\text{Im } c_{0,0,0} = 0$ , it is easy to check that the number of distinct real and imaginary parts is  $(2n + 1)^3$ . For computational reasons and after some initial experimentation, we imposed a low-order  $n = 2$  of approximation, and therefore the effective number of features, taking into account all cuboids, was  $d = 125 \cdot (2n + 1)^3 = 15,625$ .

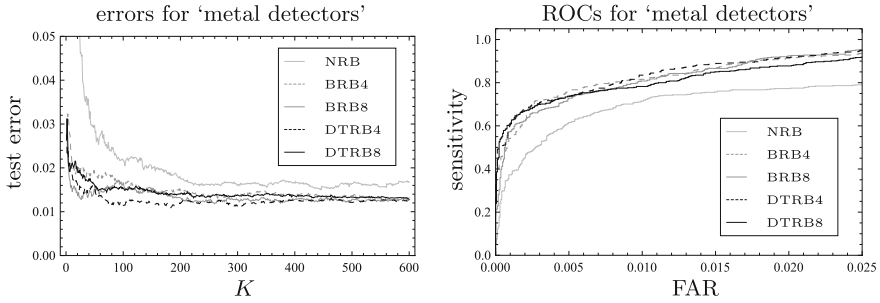
### 3.3 Results of Experiments

All results are reported with a distinction between detectors trained to detect only metal or only plastic AT mines ("Metal"/"Plastic detectors" for short).

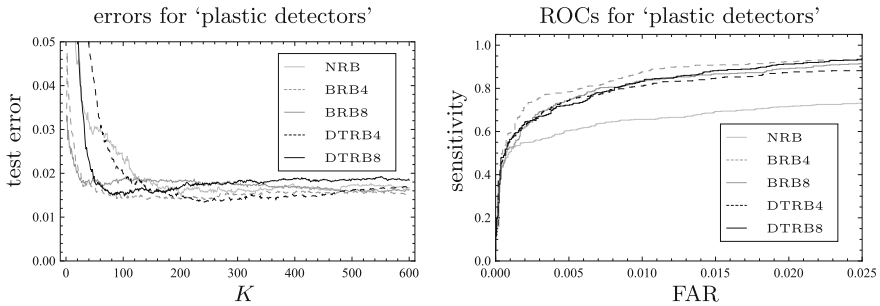
Figures 3, 4 and the Table 1 summarize results obtained at the "window-level of detail" for the 1 GB test set ( $5.5 \times 10^2$  positive windows,  $1.7 \times 10^4$  negative windows). It means that each window sampled from test C-scans is treated as separate object under detection. We remark that a positive target (mine) in the image is represented not by a single window, but a concentrated cluster of multiple windows<sup>4</sup> differing by small pixel shifts due to a certain tolerance. In the Table 1, the  $\text{AUC}_\alpha$  notation stands

<sup>3</sup> At the moment of writing this manuscript, we submitted a paper to the IEEE Trans. on Geo-science and Remote Sensing journal, proposing a technique for fast calculation of moments (11) for successive windows via multiple integral images.

<sup>4</sup> Even about 100 windows for a dense image traversal, e.g., with 1 pixel shifts.



**Fig. 3** Test errors and ROC curves for “metal detectors”



**Fig. 4** Test errors and ROC curves for “plastic detectors”

for the normalized area under ROC obtained up to the FAR equal  $\alpha$ . This shows how fast the ROC grows in its initial stage.

Results at the “image-level of detail” (for whole C-scans) are reported in the Table 2. For these results, decisions of detectors were thresholded, i.e.,  $\text{sgn}(F(\mathbf{x}) - \theta)$ , with thresholds  $\theta$  chosen from ROCs to correspond to the 0.5 sensitivity. This operation was done in order to significantly decrease the number of false alarms in images, but still keeping high detectivity. Note that if a target is represented, e.g., by a cluster of 10 windows, then the probability that at least one of them will be “turned on” (given the sensitivity 0.5 per each window) is at least  $1 - 0.5^{10} \geq 0.999$ , assuming an independence model for simplification.

The following summarizing comments on the results can be formulated. Obviously, our data set is strongly imbalanced, favoring the negative class, and so are classification results. If a zero-rule classifier is considered as a reference (accuracies 97, 96.80% respectively for metal and plastic AT mines) then all detectors surpassed it. The NRB classifier performed clearly worst, both as the “metal” and the “plastic detector.” It is particularly apparent in the ROC curves. Other classifiers exhibited only minor differences in ROCs among each other. Looking at results for the whole C-scans (see Table 2), the NRB detectors produced an unacceptably high number of false alarms (17/42 and 20/42). Both types of decision trees—DTRB4, DTRB8—turned out to be the best as “Metal detectors,” dominating the

Table 1 Test results for detectors at “windows level of detail”

Name	Acc. (%)	Sens. (%)	FAR	AUC (%)	AUC <sub>C0.01</sub> (%)	AUC <sub>C0.001</sub> (%)	FAR at Sens. 1/2	FAR at Sens. 1/3
“Metal detectors”								
Zero-rule	97.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NRB	98.32	56.50	$3.78 \times 10^{-3}$	87.62	56.61	26.49	$2.70 \times 10^{-3}$	$6.61 \times 10^{-4}$
BRB4	<b>98.76</b>	<b>65.44</b>	$2.10 \times 10^{-3}$	99.43	71.92	47.39	$7.81 \times 10^{-4}$	<b>0.00</b>
BRB8	98.68	61.75	$1.74 \times 10^{-3}$	99.46	69.20	42.72	$6.61 \times 10^{-4}$	$3.00 \times 10^{-4}$
DTRB4	98.74	61.36	$1.02 \times 10^{-3}$	<b>99.49</b>	<b>72.35</b>	52.11	$3.60 \times 10^{-4}$	<b>0.00</b>
DTRB8	98.72	59.81	<b><math>7.81 \times 10^{-4}</math></b>	99.43	71.33	<b>53.98</b>	<b><math>2.40 \times 10^{-4}</math></b>	$6.01 \times 10^{-5}$
“Plastic detectors”								
Zero-rule	96.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00
NRB	98.33	50.73	$1.02 \times 10^{-3}$	82.81	58.18	38.37	$9.00 \times 10^{-4}$	$3.00 \times 10^{-4}$
BRB4	<b>98.47</b>	<b>54.36</b>	$7.20 \times 10^{-4}$	99.30	<b>75.41</b>	<b>45.20</b>	<b><math>4.80 \times 10^{-4}</math></b>	$2.40 \times 10^{-4}$
BRB8	98.37	52.73	$1.20 \times 10^{-3}$	99.01	70.75	42.34	$7.20 \times 10^{-4}$	<b><math>1.80 \times 10^{-4}</math></b>
DTRB4	98.31	49.09	$6.60 \times 10^{-4}$	98.25	70.11	42.93	$6.60 \times 10^{-4}$	<b><math>1.80 \times 10^{-4}</math></b>
DTRB8	98.16	43.45	<b><math>3.60 \times 10^{-4}</math></b>	<b>99.40</b>	70.00	43.12	$6.60 \times 10^{-4}$	$2.40 \times 10^{-4}$

**Table 2** Test results for detectors at “images level of detail”

Object type	NRB			BRB4			BRB8			DTRB4			DTRB8		
	Detected as metal mine	Side false alarms		Detected as metal mine	Side false alarms		Detected as metal mine	Side false alarms		Detected as metal mine	Side false alarms		Detected as metal mine	Side false alarms	
“Metal detectors”															
Metal mine	11/14	7/14		14/14	1/14		13/14	0/14		14/14	1/14		13/14	0/14	
Plastic mine	1/14	3/14		0/14	2/14		0/14	1/14		0/14	0/14		0/14	0/14	
Other object	4/14	2/14		4/14	1/14		3/14	0/14		2/14	1/14		1/14	0/14	
Sens.	11/14 = 78.57%			14/14 = 100.86%			13/14 = 92.86%			14/14 = 100.0%			13/14 = 92.86%		
FAR	17/42 = 40.48%			8/42 = 19.05%			4/42 = 9.52%			3/42 = 7.14%			2/42 = 4.76%		
“Plastic detectors”															
Metal mine	3/14	6/14		1/14	3/14		0/14	0/14		1/14	3/14		0/0	4/14	
Plastic mine	13/14	0/14		12/14	4/14		13/14	1/14		13/14	0/14		13/14	1/14	
Other object	7/14	4/14		1/14	2/14		3/14	1/14		4/14	2/14		3/14	1/14	
Sens.	13/14 = 92.86%			12/14 = 85.71%			13/14 = 92.86%			13/14 = 92.86%			0/0 = 0.0%		
FAR	20/42 = 47.62%			11/42 = 26.19%			5/42 = 11.90%			10/42 = 23.81%			9/42 = 21.42%		

results of other classifiers. In particular, the DTRB4 achieved  $14/14 = 100\%$  sensitivity with  $3/42 = 7.14\%$  FAR. This can be also explained by looking at ROC/AUC characteristics—both types of trees exhibited highest  $AUC_{0.001}$ . As regards “Plastic detectors,” the results were in general slightly worse. The eight bins real boost—BRB8—surpassed other classifiers, obtaining  $13/14 = 92.86\%$  sensitivity with  $5/42 = 11.90\%$  FAR. Also, we think it is fair to remark that our material was rather difficult, containing many disruptive objects and scenes with mine resemblance. In a more natural test lane with many “empty” square meters, more appealing FAR rates could be achieved.

## 4 Conclusions

High-trust GPR systems for automatic mine detection require very large data sets to be collected and learned from. In our case study, we worked with a fairly limited collection of 210 C-scans (from  $\approx 1 \text{ m}^2$  each), but we generated a large data set from it—with approximately  $7 \times 10^4$  examples of 3D windows each described by over 15 thousand features.

Our experiments showed that a commonly applied single-feature weak learner which approximates class conditional probabilities by normal distributions (popularly known as “real boost,” NRB in our notation) was significantly outperformed by learners only slightly more refined—based on binning and shallow decision trees. In particular, the NRB yielded the AUC measures of 87.6 and 82.8% respectively for metal and plastic AT mines, whereas bins and trees (apart from one exception per eight cases) obtained AUCs greater than 99%.

## Appendix A: Notes on Boosting’s Connection to Logistic Regression

Following [3], we point out some important properties of boosting in the context of its connection to logistic regression and additive modeling.

Think of the unknown joint probability distribution (population) from which the data is drawn. Let  $p(\mathbf{x}, y)$  denote its density function, which can also be expressed as  $p(\mathbf{x}, y) = P(y|\mathbf{x})p(\mathbf{x})$ . Now, consider the exponential criterion for some classifier function  $F$  defined as an expectation taken with respect to  $p$ :

$$\begin{aligned} Q_p(F) &= \mathbb{E}_p(e^{-yF(\mathbf{x})}) = \int_{\mathbf{x}} \sum_{y \in \{-1, 1\}} e^{-yF(\mathbf{x})} p(\mathbf{x}, y) \, d\mathbf{x} \\ &= \int_{\mathbf{x}} \left( P(y=-1|\mathbf{x})e^{F(\mathbf{x})} + P(y=1|\mathbf{x})e^{-F(\mathbf{x})} \right) p(\mathbf{x}) \, d\mathbf{x}. \end{aligned} \tag{12}$$

To minimize  $Q_p$  we demand that  $\partial Q_p(F)/\partial F = 0$  (in fact it suffices to minimize the inner conditional expectation) and obtain the formula for the minimizer  $F^*(\mathbf{x}) = 1/2 \log (P(y=1|\mathbf{x})/P(y=-1|\mathbf{x}))$  being half the logit transform, typical for logistic regression. If an algorithm could somehow immediately find the perfect function  $F^*$ , then the boosting procedure could be stopped after just one round. In practice, weak learners are rough approximations of  $F^*$  and multiple rounds are needed. Solving  $F^*$  for probability leads to a form of sigmoid:

$$P(y = 1|\mathbf{x}) = e^{2F^*(\mathbf{x})} / (1 + e^{2F^*(\mathbf{x})}) = 1 / (1 + e^{-2F^*(\mathbf{x})}), \quad (13)$$

and again the similarity to logistic regression can be seen. The two minor differences are: the factor of 2 in the exponent, and the fact that in the traditional logistic regression one approximates  $F^*$  by a linear model  $F^*(\mathbf{x}) \approx a_0 + a_1 x_1 + \dots + a_n x_n$ , whereas in boosting it is a linear combination of weak learners, i.e.,  $F^*(\mathbf{x}) \approx \sum_k f_k(\mathbf{x})$ , so in general arbitrary but simple functions, each being possibly a function of all variables.

Think of the *error residuals* technique, known from regression modeling in general. In an additive model built sequentially, each successive piece of approximation “explains” some part of the target quantity and that part is subtracted from the target, so that future pieces can focus on residuals. Boosting’s reweighing scheme proceeds in an akin manner. Suppose we have fixed a partial model  $F$  and would like to update it, i.e., to have  $F := F + f$ . Recall the data-based reweighing formulas (2). Let us define their population-based counterparts:

$$Z = \int_{\mathbf{x}} \sum_{y \in \{-1, 1\}} e^{-yF(\mathbf{x})} p(\mathbf{x}, y) \, d\mathbf{x}, \quad w(\mathbf{x}, y) = e^{-yF(\mathbf{x})} p(\mathbf{x}, y) / Z. \quad (14)$$

Note that  $Z$  works as a normalizer but simultaneously is our exponential criterion value for the model so far— $Q_p(F)$ . Now, consider the criterion at  $F + f$ :

$$\begin{aligned} Q_p(F+f) &= \int_{\mathbf{x}} \sum_{y \in \{-1, 1\}} e^{-y(F(\mathbf{x})+f(\mathbf{x}))} p(\mathbf{x}, y) \, d\mathbf{x} \\ &= \int_{\mathbf{x}} \sum_{y \in \{-1, 1\}} e^{-yf(\mathbf{x})} \underbrace{e^{-yF(\mathbf{x})} p(\mathbf{x}, y) / Z}_{w(\mathbf{x}, y)} \, d\mathbf{x} \cdot Z = Q_w(f) \cdot Q_p(F). \end{aligned} \quad (15)$$

Therefore, to minimize  $Q(F + f)$  it suffices to greedily minimize  $Q_w(f)$ . The distribution  $w$  indicates which “parts” of the target quantity are already explained.

## References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks, Monterey (1984)
2. Cremer, F., et al.: Feature level fusion of polarimetric infrared and GPR data for landmine detection. In: Proceedings of EUDEM2-SCOT 2003, International Conference on Requirements and Technologies for the Detection, Removal and Neutralization of Landmines and UXO, vol. 2, pp. 638–642 (2003)
3. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *Ann. Stat.* **28**(2), 337–407 (2000)
4. Frigui, H., et al.: Context-dependent multisensor fusion and its application to land mine detection. *IEEE Trans. Geosci. Remote Sens.* **48**(6), 2528–2543 (2010)
5. Hamdi, A., Missaoui, O., Frigui, H.: An SVM classifier with HMM-based kernel for landmine detection using ground penetrating radar. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4196–4199 (2010)
6. Horng, M.H.: Texture feature coding method for texture classification. *Opt. Eng.* **42**(1), 228–238 (2002)
7. Jol, H.M.: Ground Penetrating Radar: Theory and Applications. Elsevier, Oxford (2009)
8. Kłesk, P., Godziuk, A., Kapruziak, M., Olech, B.: Landmine detection in 3D images from ground penetrating radar using Haar-like features. In: ICAISC 2013, Zakopane. Lecture Notes in Artificial Intelligence, vol. 7894, pp. 559–567. Springer, Berlin (2013)
9. Ligthart, E.E., Yarovoy, A.G., Roth, F., Ligthart, L.P.: Landmine detection in high resolution 3D GPR images. In: MIKON'04, pp. 423–426 (2004)
10. Mease, D., et al.: Boosted classification trees and class probability/quantile estimation. *J. Mach. Learn. Res.* **8**, 409–439 (2007)
11. Missaoui, O., Frigui, H., Gader, P.: Land-mine detection with ground-penetrating radar using multistream discrete hidden Markov models. *IEEE Trans. Geosci. Remote Sens.* **49**(6), 2080–2099 (2011)
12. Rasolzadeh, B., et al.: Response binning: improved weak classifiers for boosting. In: IEEE Intelligent Vehicles Symposium, pp. 344–349 (2006)
13. Schapire, R.E.: The strength of weak learnability. *Mach. Learn.* **5**, 197–227 (1990)
14. Shi, Y., et al.: Landmine detection using boosting classifiers with adaptive feature selection. In: IEEE 6th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), pp. 1–5 (2011)
15. Sun, Y., Li, J.: Adaptive learning approach to landmine detection. *IEEE Trans. Aerosp. Electron. Syst.* **41**(3), 973–985 (2005)
16. Torrione, P., Collins, L.M.: Texture features for Antitank landmine detection using ground penetrating radar. *IEEE Trans. Geosci. Remote Sens.* **45**(7), 2374–2382 (2007)
17. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2001), pp. 511–518 (2001)
18. Yarovoy, A.: Landmine and unexploded ordnance detection and classification with ground penetrating radar. In: Jol, H.M. (ed.) Ground Penetrating Radar: Theory and Applications, pp. 445–478. Elsevier, Oxford (2009)
19. Yarovoy, A.G., Kovalenko, V., Fogar, L.P.: Impact of ground clutter on buried object detection by ground penetrating radar. In: International Geoscience and Remote Sensing Symposium, pp. 755–777 (2003)

Soft Computing in Computer and Information Science

Wiliński, A.; Fray, I.E.; Pejas, J. (Eds.)

2015, XVI, 446 p. 168 illus., 15 illus. in color., Softcover

ISBN: 978-3-319-15146-5