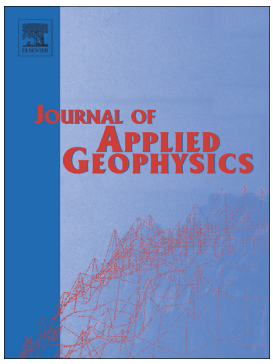

Automatic detection of pipe-flange reflections in GPR data sections using supervised learning

Pablo Bordón, Bonomo Néstor, Patricia Martinelli

PII: S0926-9851(18)30495-6

DOI: https://doi.org/10.1016/j.jappgeo.2019.103856

Reference: APPGEO 103856

To appear in: *Journal of Applied Geophysics*

Received date: 4 June 2018

Revised date: 10 September 2019

Accepted date: 11 September 2019

Please cite this article as: P. Bordón, B. Néstor and P. Martinelli, Automatic detection of pipe-flange reflections in GPR data sections using supervised learning, *Journal of Applied Geophysics*(2018), https://doi.org/10.1016/j.jappgeo.2019.103856

Automatic detection of pipe-flange reflections in GPR data sections using Supervised Learning

Bordón, Pablo[a,c], bordonpablo@gmail.com, Bonomo Néstor[a,b,*] bonomo@df.uba.ar, Martinelli, Patricia[a,b] pmartine@df.uba.ar

[a]Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Física, Buenos Aires, Argentina

[b]CONICET-Universidad de Buenos Aires. Instituto de Física de Buenos Aires (IFIBA), Buenos Aires, Argentina.

[c]On a fellowship from CONICET

[*]Corresponding author.

Abstract

Ground Penetrating radar (GPR) is a method widely used to study the near-surface subsoil. Many GPR applications require the acquisition of large volumes of data. In these cases, the processing and analysis of the data involve considerable amounts of time and human effort, and the possibility of errors increases. Considering this, the implementation of dependable methods for the automatic detection of GPR response-patterns of the targeted structures becomes clear, because they can contribute to the efficiency and reliability of the interpretation.

In this work, we present three methods for automatic detection of pipe-flange signals in constant-offset reflection-GPR images. These methods were obtained using well-known supervised machine learning techniques, and data acquired during a previous study of an extensive section of a pipeline. The first two methods are based on support vector machines (SVM), combined with the image descriptors local binary patterns (LBP) and histogram of oriented gradients (HOG), and the third, on artificial neural networks (ANN). The training and validation of these types of algorithms require large numbers of positive and negative samples. From the mentioned study, we had only 16 experimental flange-patterns. Then, in this work, they were taken as references, together with available documentation on the geometry and materials of the pipe and flanges, for building a broad database of synthetic patterns corresponding to different depths of the pipe and characteristics of the environment. These patterns constitute the set of positive samples used for training and validation. They were also used for the final test of the algorithms. The negative samples for the three stages were directly extracted from the profiles.

The results obtained indicate the usefulness of the proposed methodologies to identify the flanges. The best performance corresponded to the ANN, closely followed by SVM combined with HOG, and finally SVM with LBP. In particular, the ANN provided rates of false positive (FP) predictions for the validation and test samples of about 3%, and rates of false negative (FN) predictions of 1.67% for the validation samples and 18.75% for the test samples. Greater FN rates for the test experimental samples, in comparison to those obtained for the validation synthetic samples, were also observed for both SVM algorithms. The detection failures mainly originated in that some complex features of

the experimental flange responses could not be appropriately reproduced through the performed numerical simulations, and therefore, some of the patterns were not satisfactorily represented in the sets of positive samples used for training and validation. A first option to improve the results is to obtain a significant number and variety of experimental samples of flange responses and use them to train and validate the algorithms. Other alternatives are to use more sophisticated numerical simulation environments and to find more efficient attributes of the data.

Keywords

GPR, pipe-flange, automatic detection, SVM, ANN.

1. Introduction

GPR is one of the geophysical methods most frequently used for detecting and characterizing shallow targets in the soil (Bai and Sinfield, 2018; Barraca et al., 2016; Booth and Pringle, 2016; Jol, 2009; Qinet al., 2018). Typical targets of GPR in civil engineering and related fields of application are pipes (Carcione, 1996a; 1996b; Böniger and Tronicke, 2012; Bullo et al., 2016), reinforcement bars and different inclusions in construction materials (Stryk et al., 2013; Bonomo et al., 2015), moisture and liquid contents in materials (Orlando and Renzi, 2014; Chen et al., 2016; Ocaña-Levario et al., 2018), buried landmines and unexploded ordnance (Jin et al., 2012), archaeological structures (Bonomo et al., 2013; Azie et al., 2016) and natural layers (Okazaki et al., 2015; Gądek et al., 2016).

A relatively recent application of GPR is the detection of flanges of buried pipelines (Bonomo et al., 2011). Flanges are important points in a pipeline. Monitoring is often performed at these points to control the internal flux and to prevent leaks due to corrosion or fatigue. They are also relevant for maintenance and access to the interior of the pipelines. Nevertheless, on some occasions the positions of the flanges are unknown, as in old pipelines or when planes or signposts are missed. In these cases, it is relevant to locate them. The reflection methodology with constant-offset (CO) between antennae demonstrated very good detection capability for this type of targets. This methodology has also the advantage of not requiring altering the normal functioning of the pipelines, since it works from the ground surface in a noninvasive way.

In general, the time required for processing and analyzing the GPR data is considerably longer than the time of data acquisition. This can be especially problematic when long sections of pipelines have to be studied in the search for flanges, obtaining large amounts of data. Then, the human effort associated to the processing and analysis of the data, as well as the monetary costs become significant. In these cases, it is particularly relevant to implement efficient procedures for automatically detecting the targets, so that the interpreter can be exempted from the most strenuous tasks and focus on relevant decisions, thus minimizing the possibilities of errors.

Computer vision methodologies, which allow obtaining algorithms for the automatic detection of objects in images, have had great development during the last decades and are widely used in many different areas, such as traffic sign recognition, pedestrian detection, and gesture recognition. This type of algorithms consists of a classifier of image samples, which is the fundamental part, combined with a technique of search across the image. Their performance depends on the complexity of the objects to be recognized and their environment, as well as on a good selection and implementation of the classification algorithms and search techniques. The goal of the classifiers is to discriminate the samples that contain certain class of objects from other samples. They can be trained using supervised machine learning techniques. Ideally, the results should present invariance against the variability of the objects within each class and the changes in the environment. Common classification methods are discriminant analysis, k-nearest neighbors (k-NNs), Bayesian classifier, artificial neural networks (ANNs) and support vector machines (SVMs), among others (Bishop, 2006; James et al., 2013). Some of them work directly with the images, while others work with image attributes extracted previously by means of image feature descriptors. Two of the most extensively used descriptors are Local Binary Patterns (LBP) (Ojala et al., 1994; 2002)

and Histogram of Oriented Gradients (HOG) (Dalal et al., 2005). Regarding the search techniques, they are usually based on the concepts of pyramidal image and sliding window, and should be efficient enough as to allow quickly identifying most of the places containing the objects of interest, even if these objects have different scales.

In recent years, the use of the aforementioned methodologies for the identification of characteristic signals of different types of targets in GPR images has been increasingly frequent. In particular, ANNs have been used to detect and classify landmines and unexploded ordnance (Kobashigawa et al., 2011; Núñez-Nieto et al., 2014), to extract information about the density of asphalt pavement (Shangguan et al., 2014), to characterize inclusions in concrete (Travassos et al., 2008), to evaluate moisture and chloride contents in concrete (Sbartaï et al., 2009; Kilic and Unluturk, 2014), to estimate the thickness of remnant coal seams in mines (Strange et al., 2005), to inspect voids and conduits in hydro–electric power station tunnels (Kilic and Eren, 2018) and to locate cylindrical underground utilities (Ristić et al., 2017). On the other hand, different image descriptors have been used in combination with SVM algorithms. For example, Xie et al. (2013) used time segmentation of traces and statistical descriptors, such as the square root of variance, the standard absolute deviation and the fourth root of fourth moments, to identify air inside concrete units. Shao et al. (2011) calculated the magnitudes of the most important frequencies in radar traces to assess railway-ballast fouling conditions. Pasolli et al. (2009) calculated the amplitude spectra around the maximum of pre-located candidate reflections to classify different types of reflectors. Stone et al. (2012) and Pinar et al. (2015) used LBP, HOG and other descriptors, to locate and classify buried explosives. A fundamental advantage of these methodologies is that once a reliable algorithm for the automatic detection of a given type of target is obtained, it can be applied to detect similar targets in other surveys, provided that the characteristics of the sites are not too different.

In this paper, we present methods based on SVMs combined with HOG and LBP descriptors, and on ANNs, for the automatic detection of reflections of pipe-flanges in CO-GPR images. The experimental data used for the development of these methods correspond to the previous work by Bonomo et al. (2011). In the next two sections, we briefly describe this study and the implementation of the methods. Then, we evaluate and compare the performances obtained from each of them, and finally, we provide conclusions of the work.

2. Methodology

In general, obtaining a method for automatic detection of objects in images comprises the following stages: image formation and samples generation, attributes extraction through feature descriptors (when necessary), training of the classification algorithms by using supervised learning techniques, implementation of the search procedures, and finally, evaluation of the obtained performance.

The detection of flanges corresponds to a binary classification problem, for which the samples of the image can be separated into two classes: positive samples containing reflections at a flange, and negative samples that only contain reflections at other types of objects and noise. As usual in this kind of application, we had a small amount and variety of experimental patterns of reflections at the flanges, in comparison to what is required to train the classification algorithms. Then, simulated patterns were generated

to increase the number of positive samples. On the other hand, the negative samples were all extracted from regions of interest of the processed experimental data that did not include reflections at flanges.

The feature descriptors to be used had to provide similar results, even for samples containing reflections of flanges at different depths and in different environments. At the same time, these results had to be different from those of samples not containing reflections at flanges. We worked with the descriptors LBP (Ojala et al., 1994, 2002) and HOG (Dalal et al., 2005). They are simple and efficient, and have high discriminative power. LBP is a texture descriptor that compares the intensity of each pixel in an image to that of the surrounding pixels, whereas HOG is a shape descriptor based on the gradient of intensity of the image.

The objective of supervised learning techniques is to obtain, through the use of a set of labeled samples (training examples), classification algorithms capable of accurately predicting the class of other samples, not previously seen. Within the available methodologies, SVMs and ANNs were chosen (e.g. Bishop, 2006; James et al., 2013). The original SVM method (Vapnik and Lerner, 1963; Vapnik and Chervonenkis, 1964) was formulated for solving linear binary classification problems. In these problems, positive and negative samples, which are data points in the input space, can be separated by a hyperplane in this space. Among the infinite possible hyperplanes, linear SVM looks for the one that maximizes the separation margins between the two classes, for the training data. Later, this formulation was generalized by Cortes and Vapnik (1995) to complex non-linear problems. In these cases, the data are first projected into a higher-dimensional feature space in which they are linearly separable, using kernel functions. Then, the linear solver is applied in this space. As our classification problem was non-linear, we used this last formulation, in combination with the previous application of the LBP and HOG descriptors. For its part, ANNs are algorithms that attempt to mimic the functioning of the human brain and nervous system. These networks are able to capture subtle links between the data and discriminate into different classes, even if the underlying relationships are unknown or difficult to explain. They are composed of interconnected processing units called artificial neurons. During the training (or learning) process, the weights of the interconnections (synapses) are adjusted until a set of weights that produces an input-output mapping with minimum error is found. In particular, we used multilayer, feed-forward ANNs with backpropagation learning (e.g. Bishop, 2006), and employed architectures that did not require the application of attribute extraction procedures.

Besides the training samples, validation samples are also required to obtain an optimized classifier of a given type. First, the parameters of a number of different classifier models are adjusted using the training samples. Then, the prediction errors of these models for validation data are evaluated and, based on the results, the best ones are selected. These steps are sequentially applied until a final solution is chosen. Finally, the real error attained with this optimized classifier is evaluated using test samples, not previously employed. The described process is summarized in the workflow shown in Fig. 1. For training, validation and testing, we used the negative samples extracted from the experimental GPR profiles. Regarding the positive samples, we employed the synthetic samples for training and validation, and the experimental ones for testing.

To quantify the performance of the classifiers, confusion matrices are normally used. For binary classification problems, these matrices have the structure shown in Fig. 2, where TP and TN are the numbers of correct positive and negative predictions, respectively, and FP and FN the numbers of false positive and negative predictions, respectively. For the type of application considered in this work, it was important to reduce the number of false negative predictions, even at the expense of increasing the false positive detections, since each false negative prediction implies that a flange is not detected. In addition, the number of data belonging to the negative class (TN + FP) was much greater than that of the positive class (TP + FN). In these cases, the accuracy tends to be similar to the true negative rate (TNR), and the true positive rate (TPR) or its complement the false negative rate (FNR), are also relevant.

For the search across the GPR images, we used the conventional pyramidal image and sliding window techniques. All the algorithms were implemented using Anaconda Distribution, an open-source working environment for the Python language, and OpenCV, a free computer vision library in Python.

3. Previous GPR study

The experimental data were acquired along two sections of a mineral pipeline trajectory, 8 km long each (Bonomo et al., 2011). The pipeline is metallic, with a thin coating around it, and an external diameter d_p = 20 cm. The external diameter of the flanges is d_f = 52 cm and the length of the junction is 20 cm (Fig. 3). The surrounding soil has a predominantly sandy composition with moderate content of clay in a few places. During the prospecting, the soil water content was low, which allowed enough penetration even in the parts of the trajectory with higher clay contents. For the investigated sections, the depth to the pipeline varied in the interval 0.5 – 2.1 m, and a total of 16 flanges were identified, approximately one per kilometer.

For the survey, we used a Sensors & Software Pulse EKKO PRO system with 500 MHz antennae, which ensured enough penetration of the transmitted field and good resolution of the reflections at the flanges. A constant trace interval of 0.02 m was used, with stacking 16. The main processing steps were dewow, time-zero correction, removal of the direct waves between the antennae and application of exponential gain. Consecutive constant-offset GPR profiles were acquired along the pipeline, each one 100 m long. Propagation velocities between 11.1 and 18.8 cm/ns were obtained by fitting hyperbolae to diffraction signals in the profiles. Fig. 4 shows 4 m x 20 ns images extracted from the acquired profiles, which provide examples of the reflection patterns obtained for the pipe and the flanges. The reflections at the flanges appear with approximately hyperbolic shapes, while the reflections at the pipe have approximately flat shapes, which are not perfectly horizontal because the effect of the surface topography was not corrected. The differences that can be observed between the characteristics of the reflections in Fig. 4 are mainly due to the varying soil properties, surface topography, surrounding clutter and pipeline depth.

4. Image formation

4.1. Generation of synthetic positive samples

For the numerical simulations, we used the software Reflex-Win V.5.0.5, which calculates the fields through a finite-difference code. Among the available modeling methods, we chose the 2D exploding reflector option, as a trade-off between quality and speed. It was important that the simulated patterns were as representative as possible of those obtained experimentally, since the performance of the classifiers depends on their ability to adequately distinguish flange reflections from other signals. The simulation models represented vertical sections through the center of the pipeline (y = 0 m in Fig. 3 b), and had symmetry in the y-direction. The depth of the pipe varied in the interval 0.5 – 2.1 m. We included several irregular layers and small diffractors in the models to simulate the ground around the pipe. We used 5 - 15 layers, with relative permittivity in the interval 2.5 - 7.5 and conductivity 0.1 – 100 mS/m. As background diffractors, we used 5 – 15 circular, square and irregular reflectors, with sizes 0.05 – 0.20 m, which were located randomly. The electromagnetic parameters of the diffractors varied in the same intervals that the layers. The inclination of the air-ground interface varied in the interval -6º to 6º. 800 images were generated from the models and a positive sample of 4 m x 20 ns was extracted from each synthetic image. 560 samples were employed for training and 240 for validation. Some of these samples are shown in Fig. 5 a. They provide examples of diverse types of simulated patterns that simultaneously contain reflections at flanges, at the pipeline, and at different objects in the environment.

4.2. Extraction of negative samples from the experimental images

Negative samples are as important as positive ones, and must be representative of the background in which the target objects are embedded. First, we randomly extracted several thousands of negative samples of 4 m x 20 ns from the experimental profiles, between 6000 and 9000 depending on the choice of classification. Fig. 5 b shows examples of these samples that exhibit reflections at small diffractors, at layers with smooth interfaces, and at the pipe without flanges. Later, the selection of negative samples was refined by working with the concept of hard negative, which allows improving the performance of the classifiers by prioritizing the search for negative samples that are close to the separation boundary between classes.

Comparing the images shown in Figs. 4 and 5 b, the complexity of the problem to be solved can be appreciated. In many cases, the reflections obtained for objects that were not flanges were similar to those corresponding to flanges. The image descriptors and classifiers had to be sensitive enough to distinguish those differences. Another problem was that some of the flanges were localized in complex environments, which were difficult to reproduce in the simulations.

5. Image descriptors used with the SVM algorithms

As images were in shades of gray, the data for the LBP and HOG descriptors were the gray intensity levels of the pixels.

5.1. Local Binary Patterns

In the original formulation of the descriptor LBP, a binary number is calculated for each pixel in the image, by comparing its gray intensity to that of its 8 nearest neighbors, according to the following equation (Ojala et al., 1994):

$$\mathrm{LBP}(x,y) = \sum_{n=0}^{7} s(I_n - I) \cdot 2^n \quad (1)$$

where (x,y) are the horizontal and vertical coordinates of the pixel, I is its gray intensity, I_n for n = 0 to 7 are the gray intensities of its neighbors, which always must be numbered following the same clockwise or counterclockwise sequence, and the function s is given by:

$$s(I_n - I) = \begin{cases} 1 & \text{if } I_n - I \geq 0 \\ 0 & \text{if } I_n - I < 0 \end{cases} \quad (2)$$

This procedure is applied to all the pixels in the image, and the calculated values are stored in an output matrix (LBP image) with the same width and height as the original image.

This formulation of the operator allows capturing extremely fine details in the image, but is not suitable for capturing details at different spatial scales. Taking this into account, we used a generalization of the algorithm, circular LBP, which solves this problem (Ojala et al., 2002). The formula for circular LBP is similar to that in eqn. 1, but instead of the 8 nearest neighbors, P neighbors are considered, equally spaced on a circle of radius R, centered at (x,y). Pixel intensities are bilinearly interpolated whenever a neighbor is not at the center of a pixel. Fig. 6 shows examples of the results obtained when applying this operator on both positive and negative samples, for P = 12 and R = 3 pixels.

As can be observed, it is difficult to infer characteristics providing a clear distinction between positive and negative samples from the LBP images directly. Hence, to obtain results useful to the classifier, the image is divided into blocks of m x n pixels, and a histogram of the LBP values for the pixels in each block is calculated and stored as a vector. Often, these histograms are normalized, to account for changes of brightness and contrast across the image. Then, the normalized histograms of all the blocks are concatenated into a feature vector that is the input to the classifier, and which dimension depends on the numbers of blocks and neighbors P. The information contained in this vector changes when the parameters R and P, or the size and normalization of each block, are varied. The combinations that provide better results depend on the classifier models, and are usually sought by trial and error. We employed values of R in the range 1 – 5 pixels, values of P between 8 and 24, block sizes between 8 x 8 and 32 x 32 pixels, and L2 normalizations.

5.2. Histogram of Oriented Gradients

The HOG descriptor is especially sensitive to the shape of the objects contained in the image (Dalal et al., 2005). The general procedure for obtaining the feature vector corresponding to this descriptor is described next. First, the magnitude and orientation of the gradient for each pixel in the image are calculated. The magnitude is given by:

$$|G(x,y)| = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \quad (3)$$

where (x,y) is the position of the pixel, and G_x and G_y are respectively, the horizontal and vertical components of the gradient.

The orientation of the gradient, α, is calculated as:

$$\mathrm{tg}(\alpha(x,y)) = \frac{G_{y(x,y)}}{G_{x(x,y)}} \tag{4}$$

or:

$$\mathrm{tg}(\alpha(x,y)) = \frac{|G_y(x,y)|}{G_{x(x,y)}} \tag{5}$$

depending on whether it is "unsigned" or "signed", respectively. In the first case, the values of α range from 0 to 360 degrees, while in the second case they go from 0 to 180 degrees.

As a second step in the procedure, the image is divided into cells of size m x n pixels, and for each cell, a k_{or}–bines histogram of gradient orientations weighted by gradient magnitudes is built. Larger gradient magnitudes have greater contributions to the histogram. Each histogram corresponds to a vector of k_{or} values, which are the sums of the contributions of the pixels in the cell to each bin. Normally $k_{or} = 9$.

Instead of normalizing those histograms individually, the cells are first grouped into blocks that partially overlap, and the normalization is done based on all the histograms in each block. In Dalal and Triggs (2005), each block is composed of 2 x 2 cells, so that the blocks overlap with each other by 50%. The histograms of the four cells within a block are concatenated into a vector with 4 x k_{or} components (4 histograms x k_{or} bins per histogram) and then normalized. Finally, the normalized vectors of all the blocks are concatenated into a vector which is the HOG descriptor of the image.

In this work, we have considered unsigned gradient orientations, cell sizes between 8 x 8 and 32 x 32 pixels, and the most usual values for the rest of the options: histograms of 9 bines, and blocks of 2 x 2 cells with L2 normalization. Fig. 7 shows examples of the HOG images obtained for both positive and negative samples. The direction of each arrow indicates the dominant orientation of the gradient in the histogram of the corresponding cell. As in the case of the LBP descriptor, the direct visualization of these images does not allow to clearly differentiate the two types of samples (although it can be noted that the zones corresponding to the pipe present nearly horizontal gradients).

6. Results of the classification methods

6.1. Support Vector Machines

As explained in Section 2, the SVM method was originally formulated for solving linear binary classification problems (Vapnik and Lerner, 1963; Vapnik and Chervonenkis, 1964). Given Nt training examples of two different classes, each one consisting of a data vector x_i of dimension p, together with its corresponding label y_i (which is 1 or -1, if the datum belongs to the positive or negative classes, respectively), the goal of the

training process is to obtain a hyperplane of dimension p-1 that maximizes the separation margins between the two classes while reducing the errors in the classification.

Any hyperplane in the input space can be written as the set of points x that satisfy:

$$W \cdot x - \beta = 0 \tag{6}$$

Where W is a vector normal to the hyperplane, with norm $\|W\|$ and dimension p, and β is a constant such that $\beta/\|W\|$ is the signed distance of the hyperplane from the origin along the direction of W.

Finding the hyperplane with maximum separation margins involves the minimization of the following error function:

$$\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{Nt}\zeta_i \tag{7}$$

subject to the constraints:

$$y_i(W \cdot x_i - \beta) \geq 1 - \zeta_i \quad \text{for } 1 \leq i \leq N \tag{8}$$

The quantities ζ_i are defined by:

$$\zeta_i = \max(0, 1 - y_i(W \cdot x_i - \beta)) \quad \text{for } 1 \leq i \leq N \tag{9}$$

They are slack variables for the classification of corresponding data. ζ_i is 0 if the datum x_i lies in the correct side of the margin for its class; otherwise, ζ_i is proportional to its distance from this margin. C is a positive regularization parameter that determines the tradeoff between the maximization of the margins and the error level accepted in the classification. Higher values of C correspond to larger penalizations of the errors.

In the case of non-linear binary classification problems, the data x_i in the input space are first transformed to data $\phi(x_i)$ in a feature space of higher dimension where they are linearly separable, and then, the described linear solver is applied to these transformed data. The method is computationally efficient because this transformation is implicit; the explicit calculation of the coordinates of the transformed data is not required, only the inner products of all the pairs of these data have to be obtained. These products are defined by means of a kernel function k, as follows:

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{10}$$

This procedure, known as the kernel trick, was first introduced for SVM by Cortes and Vapnik (1995). Among the most frequently used kernels, we selected polynomial and Radial Basis Function (RBF) kernels. Polynomial kernels are given by:

$$k(x_i, x_j) = (\gamma\, x_i \cdot x_j + r)^d \tag{11}$$

where d is the degree of the polynomial kernel, and $\gamma > 0$ and $r \geq 0$ are constants. d is the most important parameter, whereas r trades off the influence of the higher-order and lower-order terms. RBF kernels are defined by:

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{12}$$

Again, γ is a constant greater than 0. Thus, kernel (12) provides a similarity measure between the two data points.

The input data were the feature vectors obtained from the LBP or HOG descriptors. According to the most common approach, the parameters of each type of kernel were optimized by grid search, together with the corresponding regularization factor C, for different values of the parameters of the descriptors LBP and HOG, in the ranges mentioned in the former section. The kernel parameters and value of C that produced the best results for each type of kernel, and their respective performances for the validation data are summarized in Table 1. The corresponding parameters of the descriptors were P = 8, R = 3, block size 16 x 16 pixels and normalization L2, for LBP, and cell size 16 x 16 pixels, cells per block 2 x 2, 9 divisions of orientation and normalization L2, for HOG.

Kernel function	Kernel Parameters and Regularization constant	HOG Accuracy	HOG False negative rate	HOG False positive rate	LBP Accuracy	LBP False negative rate	LBP False positive rate
Polynomial	d=7 γ=0.4 C=0.5	85.22%	5.82%	11.46%	79.19%	7.23%	15.74%
RBF	**γ= 0.24 C=0.1**	**96.02%**	**2.92%**	**4.10%**	**90.94%**	**2.91%**	**9.76%**

Table 1. *The two types of kernels considered, together with the kernel parameters and regularization constant that provided the best results in each case, and the corresponding performances for the validation data with the HOG and LBP descriptors. The best solution corresponded to the RBF kernel. The parameters for LBP were: P=8, R=3, block size 16 x 16 pixels and normalization L2, and for HOG: cell size 16 x 16 pixels, cells per block 4, divisions of orientation 9 and normalization L2.*

The best solution corresponded to the RBF kernel with $\gamma = 0.24$ and C = 0.1. The confusion matrices for this kernel, for validation and test data, and the LBP and HOG descriptors with the indicated parameters, are shown in Fig. 8.

For both validation and test data, the number of negative samples was much greater than the number of positive samples. Then, according to that explained in Section 2, the obtained accuracies were in the four cases similar to the TNRs, which are complementary of the false positive rates, FPRs. Considering this, we focused the analysis directly on the FPRs associated to the negative data class, and the FNRs associated to the positive data class. Regarding the FPRs, for the two descriptors, the results obtained for the validation and test data were similar. This is reasonable, considering that all those data were extracted from the experimental profiles; the lowest values corresponded to the HOG descriptor (4.10% and 4.58%, for validation and test data, respectively). As for the FNRs, the values obtained with both descriptors for the

validation data were almost coincident and very low (about 2.9%). For the test data, the FNRs were much greater: 37.5% for LBP and 25% for HOG, meaning respectively that 6 and 4 of 16 flanges were not detected. Again, the performance of the HOG descriptor was better. The samples corresponding to the 4 flanges not detected with this descriptor are shown, as an example, in Fig. 9. The probable reason of the detection failures is that some characteristics of these experimental images were not adequately reproduced by the numerical simulations used for the training and validation of the algorithms. For instance, in Fig. 9 a, very abundant and intense clutter around the pipe and flange reflections can be observed, as well as large fluctuations in the correspondent reflection traveltimes. In Fig. 9 b, the flange reflection is markedly non-symmetrical, as its right-side asymptote is not visible. Finally, in Fig. 9 c the flange reflection is significantly weaker than that of the pipe, whereas in Fig. 9 d, the pipe reflection is almost not visible to the right of the flange reflection.

6.2. Artificial Neural Networks

We used multilayer feed-forward ANNs (see e.g. Bishop, 2006). These networks consist of an input layer, one or more hidden layers and an output layer, each one composed of certain number of neurons, such that the neurons in a given layer only connect with those in the next layer. The basic operation of a neuron j in a layer L of the network is described next. First, through a propagation function f_{prop}, the outputs of the neurons i_1, …, $i_{n(L-1)}$ of the layer L-1 connected to j ($o^i_1, \dots, o^i_{n(L-1)}$, respectively) are transformed into the network input net_j, taking in consideration their connection weights with j ($\omega^{ij}_1, \dots, \omega^{ij}_{n(L-1)}$, respectively) and a bias term b_j. This term represents the weight of the connection between the neuron j and a bias neuron b, with output o_b=1.

The most common propagation function is the weighted sum:

$$f_{prop}\left(o^i_1, \dots, o^i_{n(L-1)}, \omega^{ij}_1, \dots, \omega^{ij}_{n(L-1)}, b_j\right) = \sum_{k=1}^{n(L-1)} o^i_k \omega^{ij}_k + b_j = net_j \tag{13}$$

net_j is the input to an activation function, f_{act}, also called transfer function, which determines the activation state of the neuron j, a_j

$$f_{act}(net_j) = a_j \tag{14}$$

Typical choices for f_{act} are Fermi and hyperbolic tangent functions. Finally, a_j is transformed by an output function f_{out}, which gives the output of the neuron o_j. Usually, f_{out} is the identity function, so that $o_j = a_j$.

The architecture of this type of networks is determined by the number of layers, the number of neurons per layer, the selected propagation, activation and output functions, and the learning rule. The training examples consist of data vectors of two or more classes, which are entered to the input layer, together with the expected network outputs (this is the expected values of the outputs of the neurons in the output layer). The number of neurons in the input layer is equal to the dimension of these vectors. During the training, the weights of the interconnections between the neurons and the bias terms

are optimized to minimize certain error (or loss) function, which measures the difference between the expected and the obtained results.

We trained a variety of architectures, using as learning rule the back propagation algorithm (Werbos, 1974; Rumelhart and McClelland, 1986), which is basically a gradient descent optimization procedure. In agreement with the approach most usually applied in the area, we used the propagation function given by eqn. 13, the identity function for f_{out}, and started the training of each model with random weights. The data fed to the input layer were the vectors containing the gray intensities of the pixels of the training samples. We considered sets of samples with different pixel resolutions. The output layer consisted of two neurons, one for each labeled class. Following a trial and error procedure, we varied the number of hidden layers (between 1 and 6) and the number of neurons in these layers. Then, for each one of these cases, we varied the type of gradient descent procedure, the learning rate associated to this procedure in order to reduce the possibility of falling into local minimums, the regularization technique that determines when the training is stopped, to avoid over fitting, and the activation and loss functions, to further improve convergence. In particular, we tested the performances of the optimization procedures BGD (batch gradient descent) and SGD (stochastic gradient descent), with constant and exponential learning rates, the regularization techniques early stopping and Dropout, the activation functions Sigmoid, Softmax and ReLU, and the loss functions exponential, cross entropy and L2.

Among all the tested architectures, the best performance corresponded to 4 hidden layers with 9454, 1140, 137 and 16 neurons, respectively, SGD optimization procedure with exponential learning rate, dropout regularization, Sigmoid activation function for the input and hidden layers, Softmax activation function for the output layer and cross entropy loss function. This algorithm was selected as our final ANN implementation. For the set of validation samples of 360 x 360 pixels, it provided an accuracy of 97.01%, with 3.14% of FPR and only 1.67% of FNR.

The following two tables provide a measure of how the performance of the classifiers changed when different settings were varied with respect to those of the final ANN model. In Table 2, the number of hidden layers and number of neurons per hidden layer were changed, without modifying the rest of the settings. With more than 4 hidden layers the convergence became unstable.

Number of hidden layers	Number of neurons per hidden layer	Accuracy	False negative rate	False positive rate
1	650	63.11%	29.58%	37.71%
2	2307/68	75.81%	23.33%	24.28%
3	5571/396/28	87.69%	13.33%	12.19%
4	**9454/1140/137/16**	**97.01%**	**1.67%**	**3.14%**

Table 2. *Accuracy, FNR and FPR achieved for the validation data samples of 360 x 360 pixels, with different numbers of hidden layers and neurons per layer. The rest of the settings were those corresponding to the final ANN algorithm described in the text: SGD procedure with exponential learning rate, Dropout regularization, Sigmoid activation function for the input and hidden layers, Softmax activation function for the output layer, and cross entropy loss function.*

Complementary, in Table 3, the optimization algorithm, the activation function of the output layer and the regularization technique were varied, keeping the rest of the settings unchanged.

Optimization algorithm	Activation function for the output layer	Regularization	Accuracy	False negative rate	False positive rate
BGD	ReLU	Early stopping	91.07%	6.35%	2.58%
BGD	Softmax	Dropout	94.11%	3.48%	2.41%
SGD	ReLu	Early stopping	93.86%	2.72%	3.42%
SGD	**Softmax**	**Dropout**	**97.01%**	**1.67%**	**3.14%**

Table 3. *Accuracy, FNR and FPR obtained for the validation data samples of 360 x 360 pixels, with different optimization algorithms, activation functions of the output layer, and regularization techniques. The rest of the settings are the same as in the final ANN model.*

As mentioned before, for the particular application presented in this paper, it was especially important to reduce the FNR as much as possible. As can be observed from the tables, the final ANN architecture presented the highest accuracy and lowest FNR. Although the models corresponding to the first two rows of Table 3 had lower FPRs, their FNRs were much higher.

The confusion matrices for the final ANN, for validation and test data, are shown in Fig. 10. As in the case of the SVM classifier, for the validation and test data, the number of negative samples was much greater than the number of positive samples. Then, in both cases, the accuracies obtained were similar to the TNRs. The FPRs obtained for the validation and test negative data were similar: 3.14% and 3.38%, respectively. These values are lower than those obtained for the SVM algorithm with the HOG descriptor (4.10% and 4.58%, for the validation and test negative data, respectively). Regarding the FNRs, the values obtained with the ANN (1.67% and 18.75%, for the validation and test positive data, respectively) were also better than those obtained with the SVM classifier with HOG (2.92% and 25%, respectively). Although the flanges corresponding to the samples shown in Figs. 9, a to c, remained undetected by the ANN, the one in Fig. 9 d was adequately classified.

In Fig. 11 we show an example of true and false positive detections in an experimental image, obtained with the final ANN, using pyramidal image and sliding window techniques.

7. Discussion and conclusions

The performances of the three algorithms, SVM-LBP, SVM-HOG and ANN, for the validation data were very good. They exhibited low FPRs of about 9.76%, 4.10% and 3.14%, respectively, and low FNRs of 2.9% for SVM-LBP and SVM-HOG, and 1.67% for the ANN. This indicates the usefulness of these methodologies to detect the flanges. The best results corresponded to the ANN, closely followed by SVM-HOG, which in turn was somewhat better than SVM-LBP. This is probably so because LBP is a texture descriptor, whereas HOG is especially sensitive to the shape of the objects and,

therefore, seems to be more suitable for adequately identifying the patterns of the reflections at the flanges.

Regarding the test data, the obtained FPRs were almost equal to those obtained for the validation data. This is reasonable, considering that both types of negative samples were extracted from the measured profiles. On the contrary, the FNRs were significantly higher than for the validation data: 6, 4 and 3 of the 16 experimental positive samples were misclassified by SVM-LBP, SVM-HOG, and ANN, respectively. Again, the best performance corresponded to the ANN, followed by SVM-HOG, and SVM- LBP. The probable reason for these poorer performances in comparison to those obtained for the synthetic validation positive samples is that some characteristics of the experimental flange responses could not be appropriately reproduced through the numerical simulations, so these patterns were not satisfactorily represented in the sets of synthetic samples used to train and validate the algorithms. For instance, FNs occurred in cases with profuse clutter and significant variations of the amplitude and shape of the flange and pipe reflections. The last characteristics were mainly consequences of local wave-field attenuation and deviation of the transmitted radiation lobe from the antennae-reflectors direction due to changes in the air-soil topography (e.g. due to a pit or a mound), which moreover produced fluctuations in the reflection times.

A main objective of this work was to explore the usefulness of the classification methodologies to detect the flanges. Since the number of synthetic positive samples required to train and validate the algorithms was quite high, there was a compromise between the complexity and duration of the numerical simulations and the similarity of the simulated patterns with the experimental ones. As a trade-off between quality and speed, we employed 2D models and prioritized the generation of patterns that reproduced the more frequent characteristics of the reflections. In this respect, the results obtained for the positive experimental samples with the algorithms ANN and SVM-HOG, have been satisfactory as a first approach to the subject, since only reflections with anomalous characteristics were not detected by the algorithms.

In conclusion, the potential of the two methods, ANN and SVM combined with HOG, for reliably and efficiently detecting pipe flanges in CO-GPR data-sections is promising. One advantage of the SVM algorithms is that they are much simpler and faster to implement than ANNS. In the studied case, SVM-HOG provided results almost as good as the ANN. Future work is planned to refine the algorithms to improve their ability to identify more generally complex flange patterns. For this, the sets of positive training and validation samples have to be expanded to include a wider range of possible characteristics of the experimental signals. This could be achieved by including in these sets a significant number and variety of experimental samples, by using more sophisticated numerical modeling, and/or by calculating other types of data attributes.

Acknowledgements

This work was partially supported by ANPCyT, PICT-2014-2285, and CONICET.

References

Azie, S., Aziz, R., Stewart, R., Green, S.L. and Flores, J.B., 2016. Locating and characterizing burials using 3D ground-penetrating radar (GPR) and terrestrial laser

scanning (TLS) at the historic Mueschke Cemetery, Houston, Texas. Journal of Archaeological Science: Reports, 8, 392–405.

Bai, H., Sinfield, J., 2018. Effects of GPR antenna configuration on subpavement drain detection based on the frequency-shift phenomenon. Journal of Applied Geophysics 146, 198–207.

Barraca, N., Almeida, M., Varumc, H., Almeida, F., Senos Matias, M., 2016. A case study of the use of GPR for rehabilitation of a classified Art Deco building: The InovaDomus house. Journal of Applied Geophysics 127, 1–13.

Bishop, C., 2006. Pattern recognition and machine learning. Springer-Verlag, New York, pp. 738.

Booth, A. and Pringle, J., 2016. . Semblance analysis to assess GPR data from a five-year forensic study of simulated clandestine graves. Journal of Applied Geophysics 125, 37–44.

Böniger, U. and Tronicke, J., 2012. Subsurface utility extraction and characterization: Combining GPR symmetry and polarization attributes. IEEE Transactions on Geoscience and Remote Sensing, 50, 736-746.

Bonomo, N., de la Vega, M., Martinelli, P. and Osella, A., 2011. Pipe-flange detection with GPR. Journal of Geophysics and Engineering, 8, 35–45.

Bonomo, N., Osella, A. and Ratto, N., 2013. GPR investigations at an Inca-Spanish site in Argentina. Near Surface Geophysics, 11, 449-456.

Bonomo, N., Bullo, D., Villela, A. and Osella, A., 2015. Ground-penetrating radar investigation of the cylindrical pedestal of a monument. Journal of Applied Geophysics, 113, 1–13.

Bullo, D., Villela, A. and Bonomo, N., 2016. Azimuth calculation for buried pipelines using a synthetic array of emitters, a single survey line and scattering matrix formalism. Journal of Applied Geophysics, 134, 253–266.

Carcione, J. M., 1996a, Ground radar simulation for archaeological applications. Geophysical Prospecting, 44, 871-888.

Carcione, J. M., 1996b, Ground-radar numerical modelling applied to engineering problems. European Journal of Environmental and Engineering Geophysics, 1, 65-81.

Chen, D.H., Scullion, T. and Nam, B.H., 2016. Characterization of structural conditions for pavement rehabilitations. Construction and Building Materials, 121, 664–675.
Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine Learning, 20, 273–297.

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1, 886-893.

Fawcett, T., 2006. An Introduction to ROC Analysis. Pattern Recognition Letters. 27, 861–874.

Gądek, B., Grabiec, M., Kędzia, S. and Rączkowska, Z., 2016. Reflection of climate changes in the structure and morphodynamics of talus slopes (the Tatra Mountains, Poland). Geomorphology, 263, 39–49.

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning, with applications in R. Springer-Verlag, New York, pp. 426.

Jin, T., Lou, J. and Zhou, Z., 2012. Extraction of landmine features using a forward-looking Ground-Penetrating Radar with MIMO array. IEEE Transactions on Geoscience and Remote Sensing, 50(10), 4135-4144.

Jol H.M., 2009. Ground Penetrating Radar theory and applications. Elsevier Science, Amsterdam, pp. 509.

Kilic, G. and Unluturk, M.S., 2014. Performance evaluation of the neural networks for moisture detection using GPR. Nondestructive Testing and Evaluation, 29(4), 283–296.

Kilic, G. and Eren, L., 2018. Neural network based inspection of voids and karst conduits in hydro–electric power station tunnels using GPR. Journal of Applied Geophysics 151, 194–204.

Kobashigawa, J.S., Youn, H.S., Iskander, M.F. and Yun, Z., 2011. Classification of buried targets using Ground Penetrating Radar: comparison between Genetic Programming and Neural Networks. IEEE Antennas and Wireless Propagation Letters, 10, 971-974.

Núñez-Nieto, X., Solla, M., Gómez-Pérez, P. and Lorenzo, H., 2014. GPR signal characterization for automated landmine and UXO detection based on Machine Learning Techniques. Remote Sensing, 6, 9729-9748.

Ocaña-Levario, S., Carreño-Alvarado, E., Ayala-Cabrera, D. and Izquierdo, J., 2018. GPR image analysis to locate water leaks from buried pipes by applying variance filters. Journal of Applied Geophysics 152, 236–247.

Ojala, T., Pietikäinen, M. and Harwood, D., 1994. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1, 582 - 585.

Ojala, T., Pietikäinen, M. and Mäenpää, T., 2002. Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(7), 971-987.

Okazaki, H., Kwak, Y. and Tamura, T., 2015. Depositional and erosional architectures of gravelly braid bar formed by a flood in the Abe River, central Japan, inferred from a three-dimensional ground-penetrating radar analysis. Sedimentary Geology, 324, 32–46.

Orlando, L. and Renzi, B., 2014. Electrical permittivity and resistivity time lapses of multiphase DNAPLs in a lab test. Water Resources Research, 51, 377–389.

Pasolli, E., Melgani, F. and Donelli, M., 2009. Automatic analysis of GPR images: a pattern-recognition approach. IEEE Transactions on Geoscience and Remote Sensing, 47, 2206-2217.

Pinar, A., Masarik, M., Havens, T.C., Burns, J., Thelen, B. and Becker, J., 2015. Approach to explosive hazard detection using sensor fusion and multiple kernel learning with downward-looking GPR and EMI sensor data. Proceedings of SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX, 94540B.

Qin, T., Yonghui, Lin, G., Hu, S., An, C., Geng, D. and Rao, C., 2018. Underwater archaeological investigation using ground penetrating radar: A case analysis of Shanglinhu Yue Kiln sites (China). Journal of Applied Geophysics 154, 11–19.

Ristić, A., Bugarinović, Ž., Vrtunski, M. and Govedarica, M., 2017. Point coordinates extraction from localized hyperbolic reflections in GPR data. Journal of Applied Geophysics 144, 1–17.

Rumelhart, D.E. and McClelland, J.L., 1986. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 2: psychological and biological models. MIT Press, Cambridge, MA, USA, pp 611.

Sbartaï, Z.M., Laurens, S., Viriyametanont, K., Balayssac, J.P. and Arliguie, G., 2009. Non-destructive evaluation of concrete physical condition using radar and artificial neural networks. Construction and Building Materials, 23, 837–845.

Shangguan, P., Al-Qadi, I.L. and Lahouar, S., 2014. Pattern recognition algorithms for density estimation of asphalt pavement during compaction: a simulation study. Journal of Applied Geophysics, 107, 8–15.

Shao, W., Bouzerdoum, A., Su, L., Indraratna, B. and Rujikiatkamjorn, C., 2011. Automatic classification of Ground-Penetrating-Radar signals for railway-ballast assessment. IEEE Transactions on Geoscience and Remote Sensing, 49, 3961-3972.

Stone, K., Keller, J., Anderson, D. and Barclay, D., 2012. An automatic detection system for buried explosive hazards in FL-LWIR and FL-GPR data. Proceedings of SPIE 8357, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII, 83571E.

Strange, A.D., Ralston, J.C. and Chandran, V., 2005. Near-surface interface detection for coal mining applications using Bispectral Features and GPR. Subsurface Sensing Technologies and Applications, 6(2), 125–149.

Stryk, J., Matula, R. and Pospisil, K., 2013. Possibilities of ground penetrating radar usage within acceptance tests of rigid pavements. Journal of Applied Geophysics, 97, 11–26.

Travassos, X.L., Vieira, D.A.G., Ida, N., Vollaire, C. and Nicolas, A., 2008. Characterization of inclusions in a nonhomogeneous GPR problem by Artificial Neural Networks. IEEE Transactions on Magnetics, 44(6), 1630-1633.

Vapnik, V. and Lerner, A., 1963. Pattern recognition using generalized portrait method. Automation and Remote Control, 24, 774–780.

Vapnik, V. and Chervonenkis, A., 1964. On a class of perceptrons. Automation and Remote Control, 25(1), 103-109.

Werbos, P.J., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD Thesis, Harvard University, Cambridge, MA, USA, pp. 453.

Xie, X., Li, P., Qin, H., Liu, L. and Nobes, D.C., 2013. GPR identification of voids inside concrete based on the support vector machine algorithm. Journal of Geophysics and Engineering, 10, 1-10.

Figure captions

Fig. 1. Workflow of the process of obtaining the classifier algorithms.

Fig. 2. Confusion matrix for a binary classification problem (Fawcett, 2006), where TP and TN are the numbers of true positive and negative predictions, respectively, and FP and FN are the numbers of false positive and negative predictions, respectively.

Fig. 3. a) Photograph and b) schematic model of the investigated pipes and flanges.

Fig. 4. 4 m x 20 ns images extracted from the experimental CO-GPR profiles, which exemplify the typical reflections at the pipe (red arrows) and at the flanges (yellow arrows). Surface topography is not corrected.

Fig. 5. Examples of the samples used for the implementation of the classification algorithms. All of them have dimensions 4 m x 20 ns. a) Positive samples obtained from synthetic modeling, b) negative samples extracted from the experimental profiles.

Fig. 6. a) Synthetic positive samples (left) of dimensions 4 m x 20 ns (448 x 448 pixels), together with their corresponding circular LBP images (right) obtained considering P = 12 and R = 3 pixels. b) The same as in a), but for experimental negative samples.

Fig. 7. a) Synthetic positive samples (left) of dimensions 4 m x 20 ns (448 x 448 pixels), together with their corresponding HOG images (right) obtained considering cells of 32 x 32 pixels, cells per block 4, 9 divisions of orientation and normalization L2. (b) The same as in (a), but for experimental negative samples.

Fig. 8. Confusion matrices corresponding to the RBF kernel with $\gamma = 0.24$ and $C = 0.1$, for validation and test data, and the LBP and HOG descriptors with the parameters indicated in Table 1. The structure of these matrices is shown in Fig. 2.

Fig. 9. 4 m x 20 ns samples of the flanges that were not detected with the RBF kernel with $\gamma = 0.24$ and C = 0.1, and the HOG descriptor with the parameters indicated in Table 1.

Fig. 10. Confusion matrices obtained for validation and test data samples of 360 x 360 pixels, with the final ANN architecture. The structure of these matrices is shown in Fig. 2.

Fig. 11. Example of true positive (TP) and false positive (FP) detections in a profile, obtained with the final ANN architecture, using sliding window and pyramidal image techniques. Only 40 m of the original 100 m-long profile are shown, for a better visualization of the pipe and flange signals. Different horizontal artifacts are visible in the image, due to the basic filtering procedure applied to the data.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

- Automatic detection of pipe-flange signals in simple-offset GPR images is explored.
- Support vector machines is combined with local binary patterns and histogram of oriented gradients
- Artificial neural networks are also evaluated
- Synthetic sections are used as positive samples in training and validation
- Experimental data are used as negative samples in training and validation, and also in testing
- The capabilities of these methods for detecting pipe flanges is shown

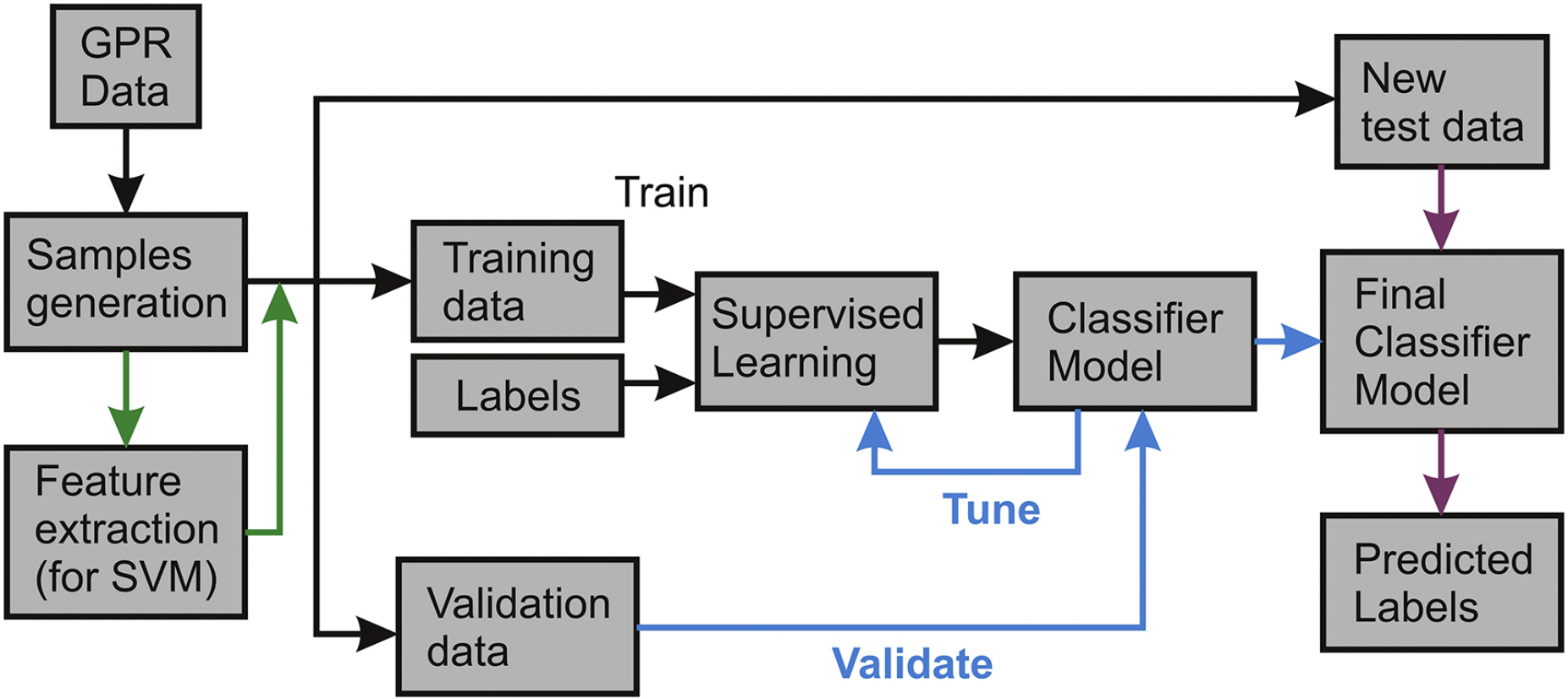
GPR Data
Samples generation
Feature extraction (for SVM)
Train
Training data
Labels
Supervised Learning
Classifier Model
Tune
Validation data
Validate
New test data
Final Classifier Model
Predicted Labels

Figure 1

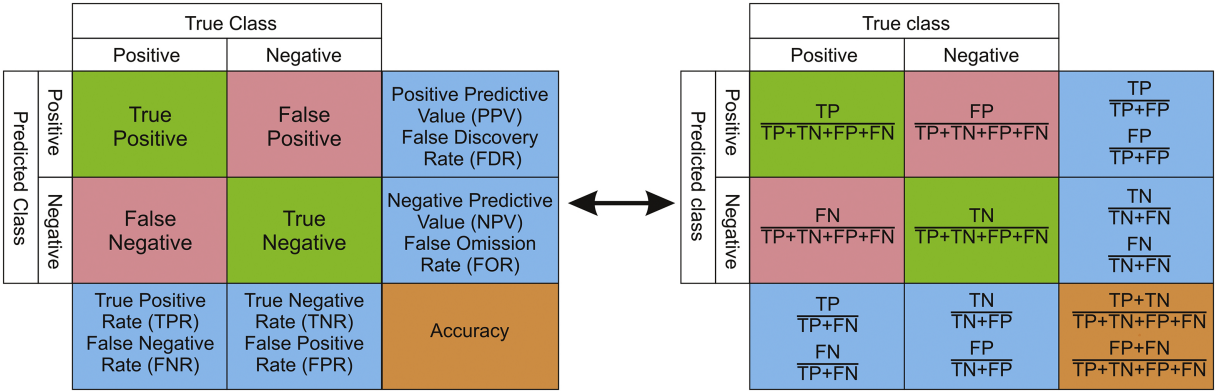
True Class
Positive
Negative
Predicted Class
Positive
Negative
True Positive
False Positive
Positive Predictive Value (PPV) False Discovery Rate (FDR)
False Negative
True Negative
Negative Predictive Value (NPV) False Omission Rate (FOR)
True Positive Rate (TPR) False Negative Rate (FNR)
True Negative Rate (TNR) False Positive Rate (FPR)
Accuracy
True class
Positive
Negative
Predicted class
Positive
Negative
TP / TP+TN+FP+FN
FP / TP+TN+FP+FN
TP / TP+FP
FP / TP+FP
FN / TP+TN+FP+FN
TN / TP+TN+FP+FN
TN / TN+FN
FN / TN+FN
TP / TP+FN
FN / TP+FN
TN / TN+FP
FP / TN+FP
TP+TN / TP+TN+FP+FN
FP+FN / TP+TN+FP+FN

Figure 2

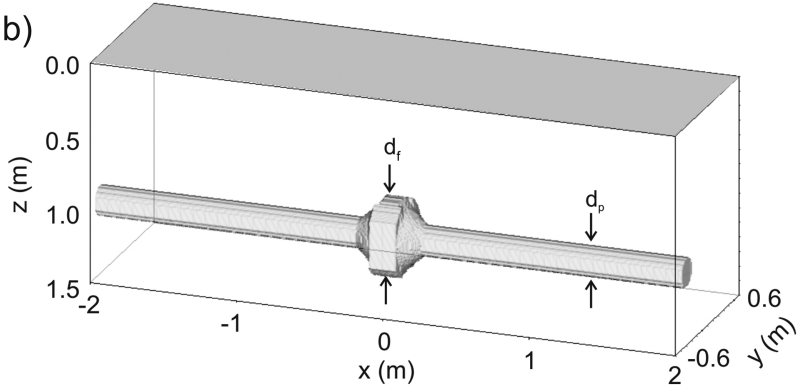

Figure 3

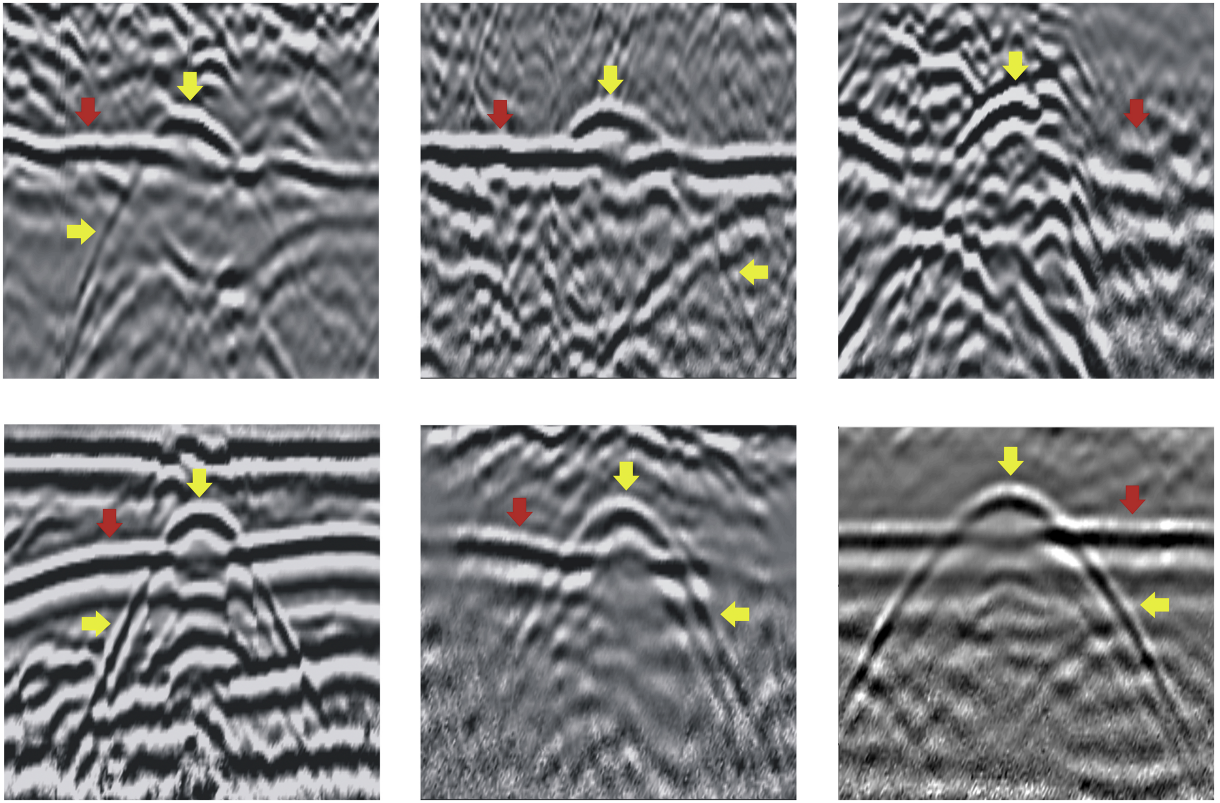

Figure 4

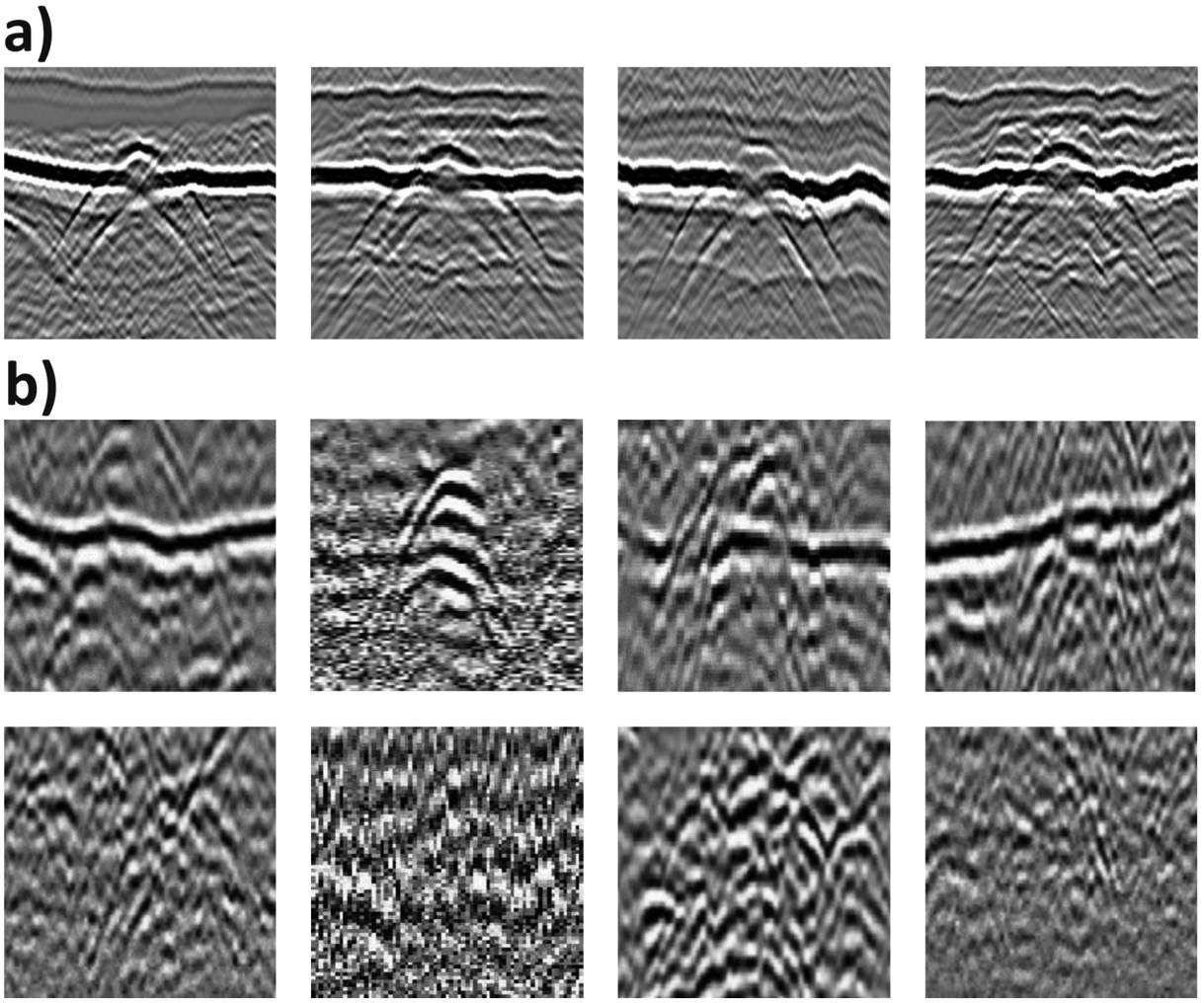

Figure 5

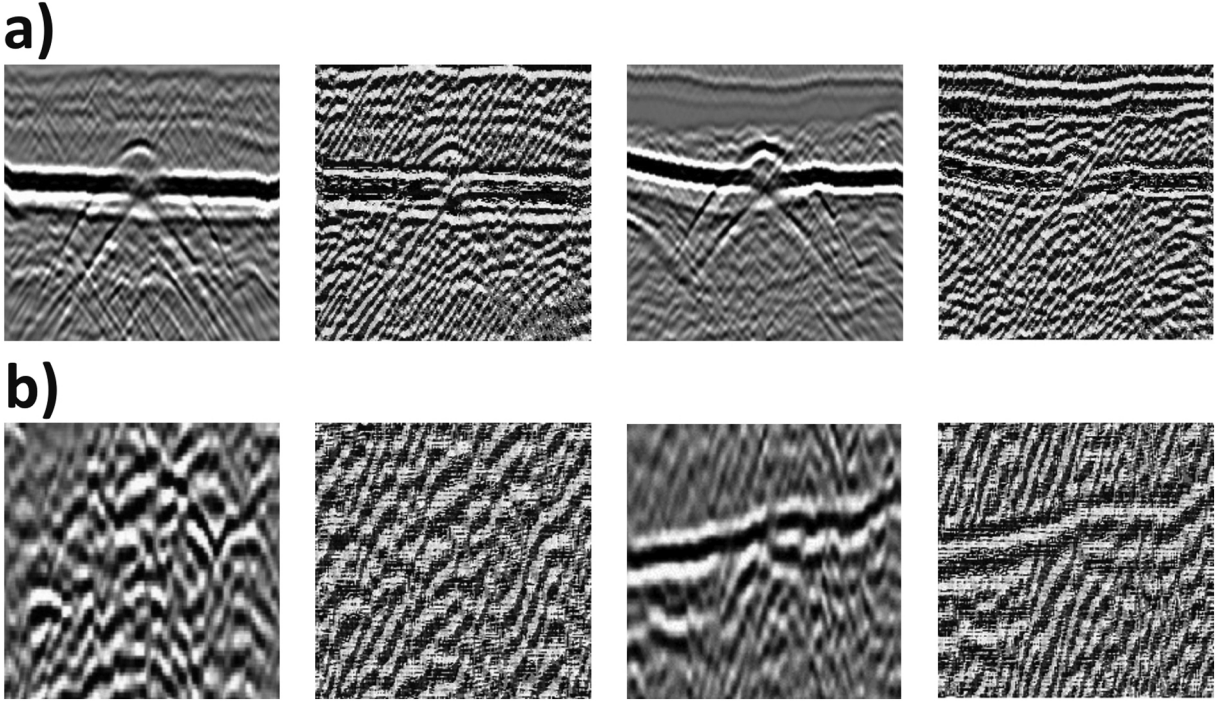

Figure 6

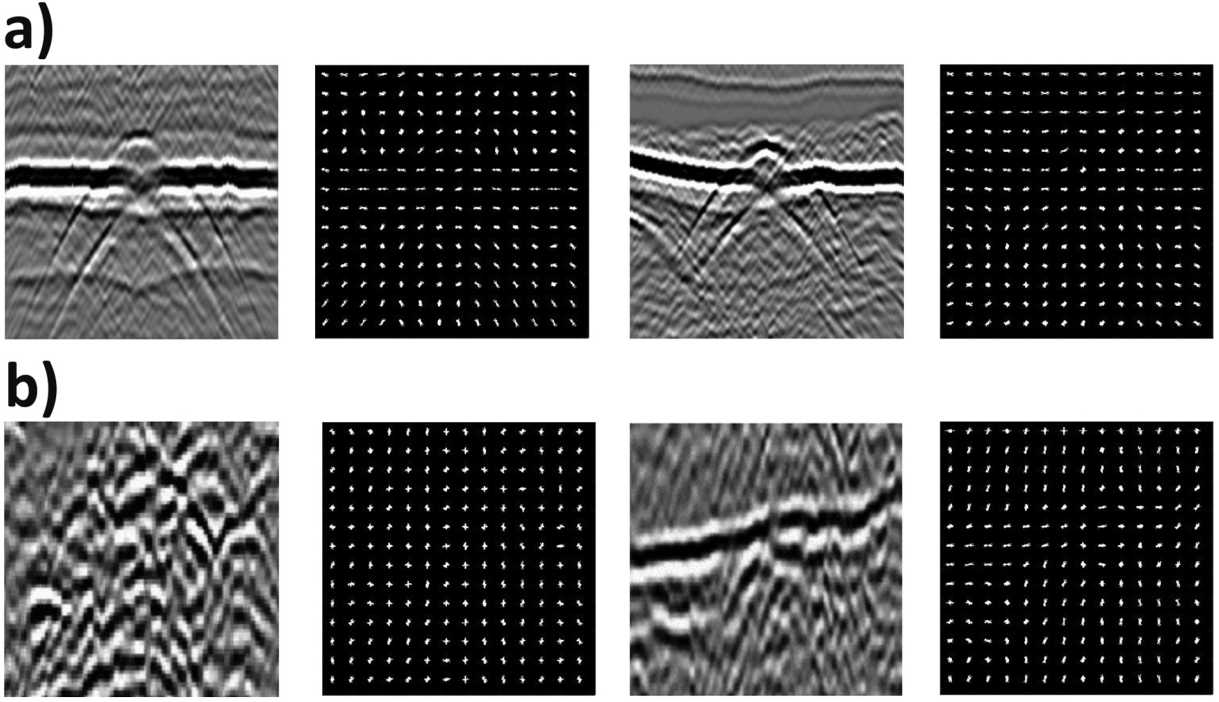

Figure 7

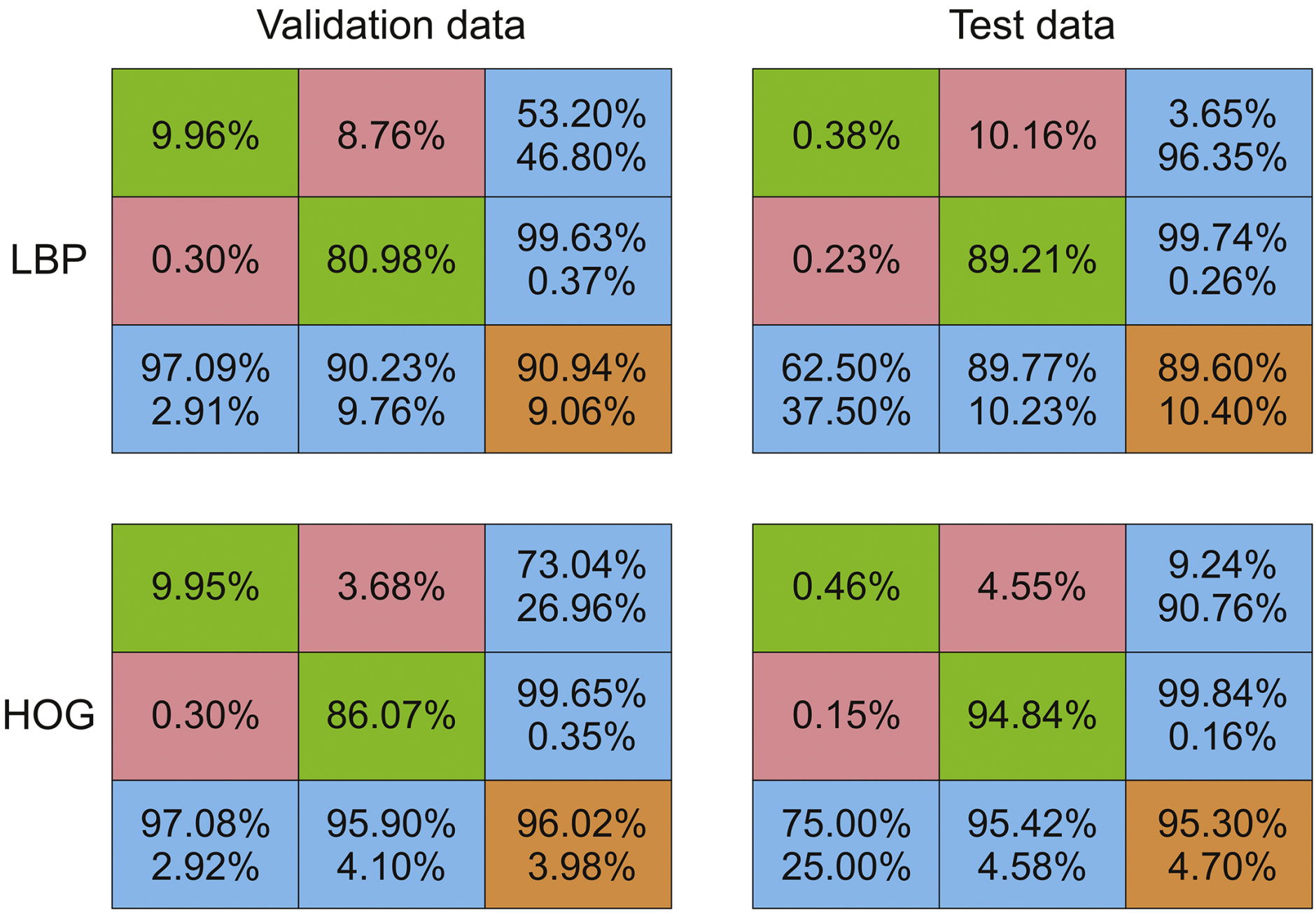
Validation data
Test data
LBP
9.96%
8.76%
53.20%
46.80%
0.30%
80.98%
99.63%
0.37%
97.09%
2.91%
90.23%
9.76%
90.94%
9.06%
0.38%
10.16%
3.65%
96.35%
0.23%
89.21%
99.74%
0.26%
62.50%
37.50%
89.77%
10.23%
89.60%
10.40%
HOG
9.95%
3.68%
73.04%
26.96%
0.30%
86.07%
99.65%
0.35%
97.08%
2.92%
95.90%
4.10%
96.02%
3.98%
0.46%
4.55%
9.24%
90.76%
0.15%
94.84%
99.84%
0.16%
75.00%
25.00%
95.42%
4.58%
95.30%
4.70%

Figure 8

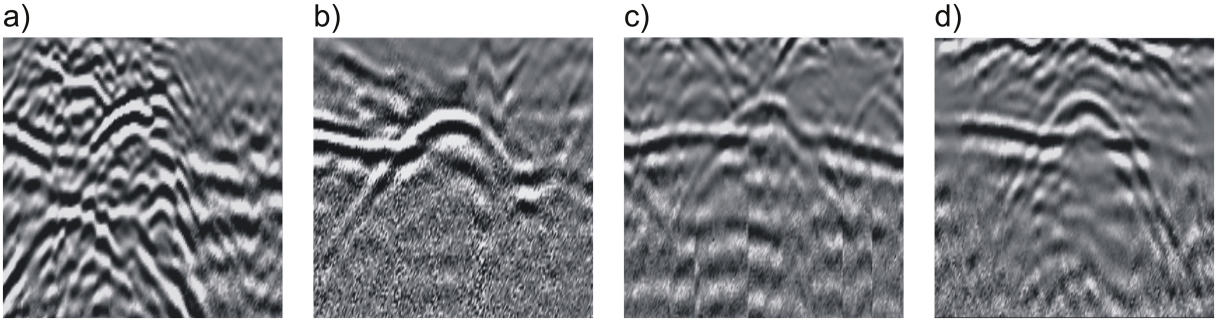

Figure 9

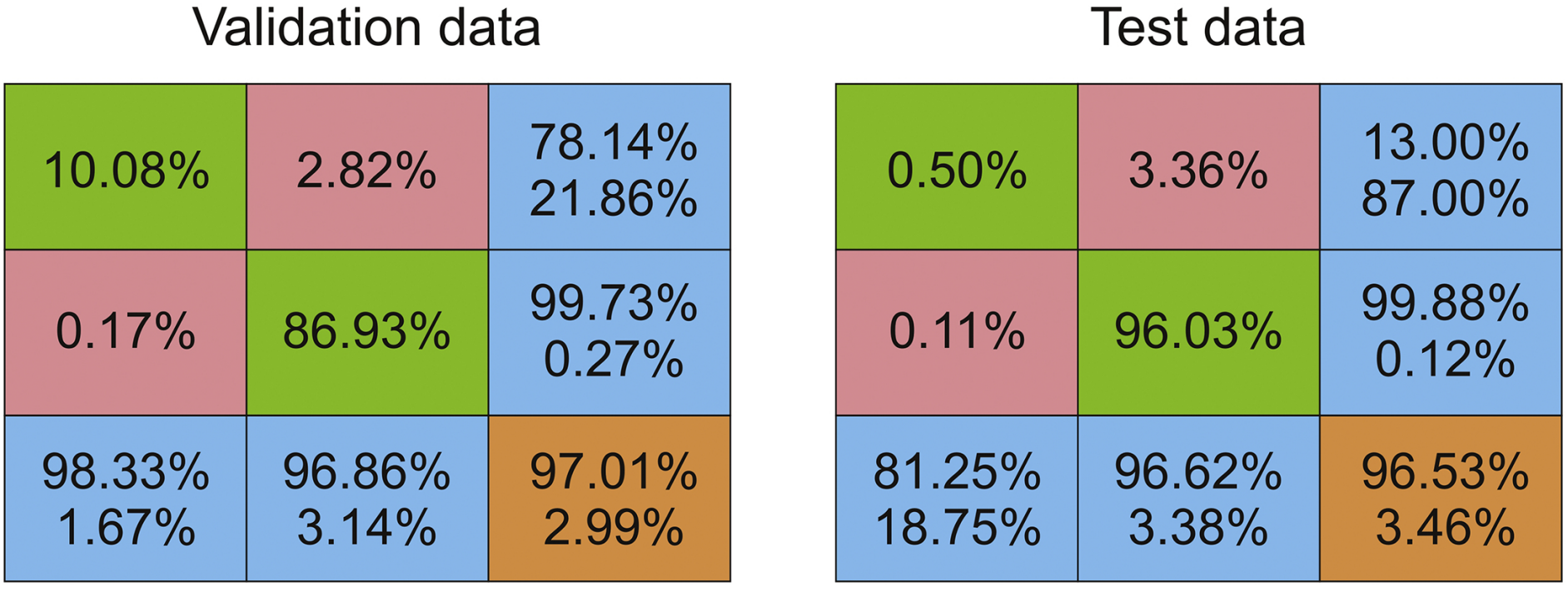
Validation data
10.08%
2.82%
78.14%
21.86%
0.17%
86.93%
99.73%
0.27%
98.33%
1.67%
96.86%
3.14%
97.01%
2.99%
Test data
0.50%
3.36%
13.00%
87.00%
0.11%
96.03%
99.88%
0.12%
81.25%
18.75%
96.62%
3.38%
96.53%
3.46%

Figure 10

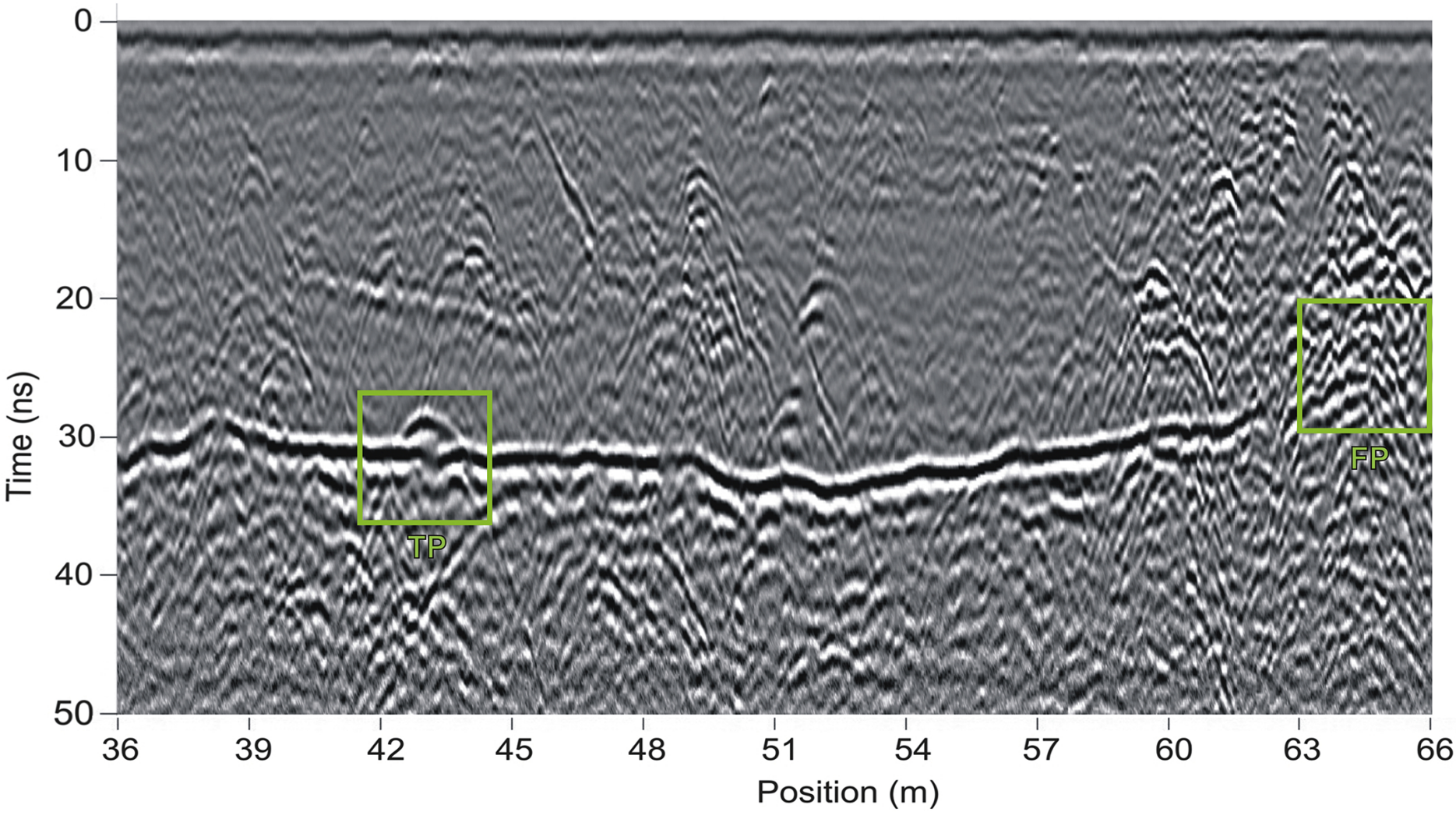
0
10
20
30
40
50
Time (ns)
36
39
42
45
48
51
54
57
60
63
66
Position (m)
TP
FP

Figure 11