

BURIED OBJECT DETECTION FROM B-SCAN GROUND PENETRATING RADAR DATA USING FASTER-RCNN

Minh-Tan Pham, Sébastien Lefèvre

Université Bretagne Sud - IRISA, UMR 6074, F-56000 Vannes - France
{minh-tan.pham, sebastien.lefevre}@irisa.fr

ABSTRACT

In this paper, we adapt the Faster-RCNN framework for the detection of underground buried objects (i.e. hyperbola reflections) in B-scan ground penetrating radar (GPR) images. Due to the lack of real data for training, we propose to incorporate more simulated radargrams generated from different configurations using the gprMax toolbox. Our designed CNN is first pre-trained on the grayscale Cifar-10 database. Then, the Faster-RCNN framework based on the pre-trained CNN is trained and fine-tuned on both real and simulated GPR data. Preliminary detection results show that the proposed technique can provide significant improvements compared to classical computer vision methods and hence becomes quite promising to deal with this kind of specific GPR data even with few training samples.

Index Terms— Ground penetrating radar (GPR), object detection, deep learning, Faster-RCNN

1. INTRODUCTION

Ground penetrating radar (GPR) is one of the most widely used geophysical techniques applied to detect underground buried objects such as landmines, pipes or archaeological artifacts, etc. A GPR system transmits an electromagnetic wave into the ground at several spatial positions and receives the reflected signal to form the subsurface 2-D high resolution image (a B-scan radargram). Within such images, underground objects appear particularly as hyperbolic-shaped signatures. The detection of buried objects can be therefore considered as the detection of reflected hyperbolas in GPR images, which has been tackled using image recognition and computer vision techniques so far in the literature.

Different unsupervised and supervised approaches have been investigated to perform automatic detection of buried objects using GPR B-scan images. One of the most popular and classical approaches is the Hough transform (HT) method. In [1, 2], the generalized HT and the randomized HT were employed to find the hyperbola parameters which are recorded within the Hough accumulator space. However,

most of these Hough-based approaches are still limited to the fact that the handling and discretization of a great number of parameters could lead to a huge computational complexity. In [3,4], the template matching and the dictionary-based techniques were used to determine hyperbola signatures. These methods work based on the correlation score between each GPR image patch and the template or the dictionary model. They again require many parameters for the setup and definition of different templates or dictionary models. Some other methods were proposed using supervised pattern recognition approach such as the HOG features [5], the Viola-Jones learning algorithm based on Haar-like features [6]. However, their results still involve several unexpected false alarms and missed detection targets. Thus, the detection of hyperbolas from GPR images using these classical computer vision strategies is still considered as a challenging task.

Recently, the incredible development of deep convolutional neural networks (CNNs) in computer vision domain has provided a lot of tools and frameworks to tackle various tasks in image understanding and recognition. Efforts have been done in GPR image processing. In [7], deep CNNs were exploited for classifying B-scan profiles into threat and non-threat classes. In [8], landmine detection from GPR data using CNNs has provided quite promising results compared to other vision methods. In [9], the authors discussed some good practices when applying CNNs to the detection of buried threats from GPR data. However, all of the deep learning based techniques in the literature have focused on classification or patch-based detection using sliding window over the whole image. In this work, we would like to perform an end-to-end framework for hyperbola detection from GPR B-scan. To do this, we apply the Faster-RCNN framework [10] which has proved very effective performance in computer vision domain. The contributions of the paper are follows: 1) we first define and pre-train a CNN using the grayscale Cifar-10 database and then transfer the weights into the Faster-RCNN framework; 2) the training data are created partly from the real collected GPR acquisitions and partly from the simulated radargrams obtained by gprMax toolbox [11]; 3) we train and fine-tune the Faster-RCNN based on the pre-trained weights and test on both simulated and real data in order to prove the effectiveness of the proposed approach.

The remainder of this paper is organized as follows. Section 2 first presents the real collected GPR data as well as the simulated radargrams used in this work. In Section 3, we describe the proposed approach to adapt the Faster-RCNN for hyperbola detection from our data. We then provide some preliminary detection results in Section 4. Finally, Section 5 concludes the paper and discuss some on-going as well as future perspectives.

2. DATA SETS

2.1. Real collected data

In this work, we exploit about 100 real B-scan data recently collected from several sites in Rennes, France using a GPR antenna of 300Hz. Each acquisition has a time range of 100ns in depth and can penetrate up to 7 meter. Within these data, hyperbola signatures were recorded from the reflection of electromagnetic signal on buried objects with different shapes and materials. Only some are clear and well-shaped, while most of them are weakly contrasted, asymmetric and perturbed by noises and clutters caused by the soil heterogeneity and the change of impedance between different subsurface layers. Figure 1 shows two examples of real data from which we can observe some hyperbolas of various sizes that suffer a lot of noises and clutters.

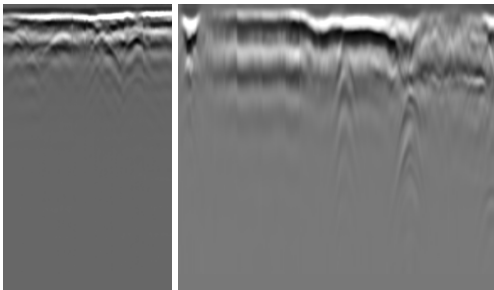


Fig. 1. Examples of real GPR radargrams.

2.2. Simulated data

Due to the lack of real data for training, more GPR radargrams were generated to simulate different scenarios using the gprMax toolbox [11]. Various configurations were considered where objects of different sizes and materials were placed at different positions and depths. The same antenna frequency of 300Hz and same time range of 100ns were set for simulation. The simulated images were then added some noises which were estimated from the real data. Figure 2 shows two examples of simulated radargrams where hyperbolas (generally well-shaped) are intersected and have crossing prongs. In this work, 50 simulated radargrams were simulated and combined with the previous real data to perform and evaluate the proposed framework.

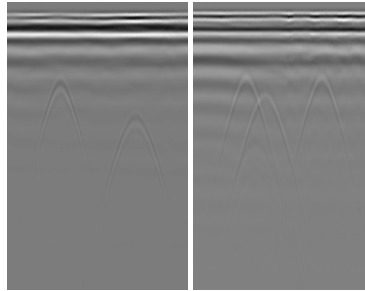


Fig. 2. Examples of simulated GPR B-scans using gprMax.

3. APPLICATION OF FASTER-RCNN

The proposed approach consists of two main stages which can be observed from Figure 3: 1) pre-train a designed CNN on the grayscale Cifar-10 database; 2) train and fine-tune the Faster-RCNN (based on pre-trained CNN weights) using both real and simulated GPR data. We now describe each of them in details.

3.1. Pre-training a CNN on the Cifar-10 database

CNNs are generally comprised of convolution layers followed by pooling layers and fully-connected layers. Our defined CNN simply includes 3 convolution layers of 16, 32 and 64 filters of size 5×5 pixels (each one is followed by a ReLu activation and a max-pooling layer of size 2×2 pixels) and one fully-connected layer of 64 neurons. As recommended in [9], the Cifar-10 was chosen (instead of ImageNet) since the image size is small (32×32 pixels), which approximates the size of hyperbolas within the studied GPR images. It can be trained faster and easier on a personal computer with limited GPU. Also, in order to match single-channel GPR data, it is better to train the CNN on grayscale Cifar-10 instead of the color database.

3.2. Training the Faster-RCNN on both real and simulated GPR data

The Faster-RCNN involves two main components: the Region proposal network (RPN) and the Fast-RCNN [10]. For a brief description, the role of RPN is to generate a set of region proposals while the Fast-RCNN (including a classifier and a box regression operator) detects whether each region is an object or not. They both share the same weights from the previously pre-trained CNN. For more details about the Faster-RCNN framework, readers are referred to [10].

As shown in Figure 3, both real and simulated GPR images were exploited to train the Faster-RCNN using the pre-trained CNN weights. All implementations in this work were carried out based on the MATLAB Neural Network toolbox using a PC with a GPU compute capacity 5.0.

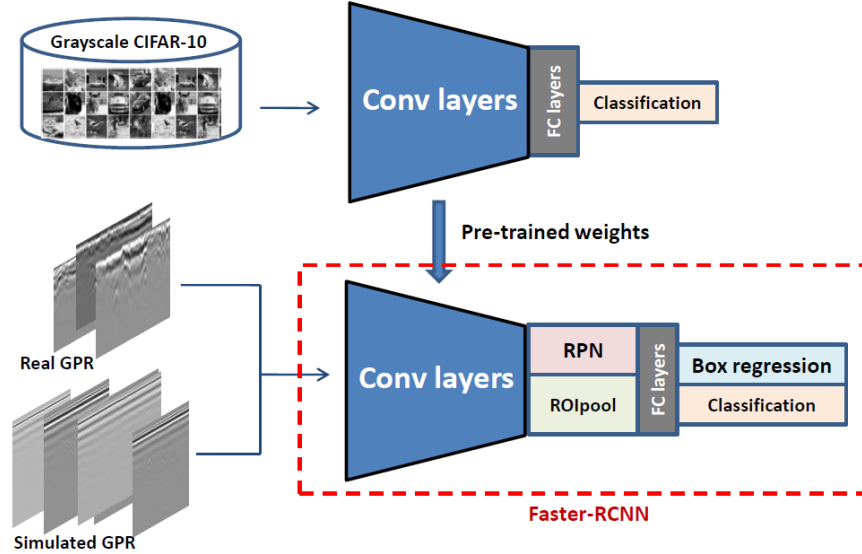


Fig. 3. Proposed framework for buried object detection from GPR B-scan images.

4. PRELIMINARY RESULTS

In order to evaluate the proposed approach on both simulated and real GPR data, three scenarios were tested:

1. Train and test on simulated data;
2. Train and test on real data;
3. Train on simulated + real data, test on real data.

We note that training and test samples were split so that they were well separated. For real radargrams, we used 60 radargrams for training and 40 for testing. For simulated data, 40 were set for training and 10 for testing. We now provide some preliminary results with qualitative assessment on each test scenario in order to confirm the effectiveness of the proposed framework.

4.1. Performance on simulated data (scenario 1)

The first scenario aims at quickly evaluating the proposed approach only on simulated data. Some detection results are reported in Figure 4 where hyperbolas were detected with high confidence scores. In general, our experiments showed that the framework can cope well with this scenario in order to provide good performance on simulated images.

4.2. Performance on real data (scenarios 2 and 3)

The other two scenarios were experimented when working on real data. The motivation is to prove that by adding more simulated radargrams for training, the detection framework could provide better performance compared to the case that

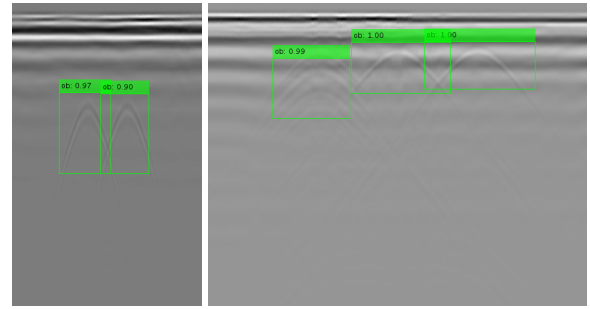


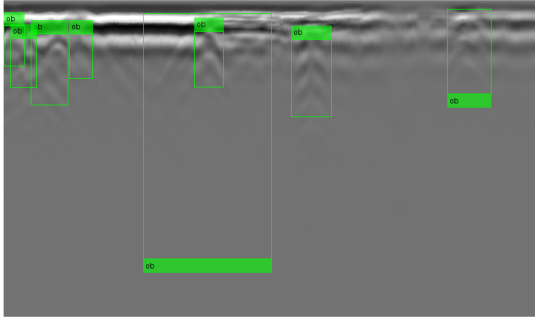
Fig. 4. Detection results on simulated data.

only real data (with limited quantity) were exploited. In addition, one classical recognition technique called cascade object detector (COD) based on the Viola-Jones algorithm [12] was implemented for comparison.

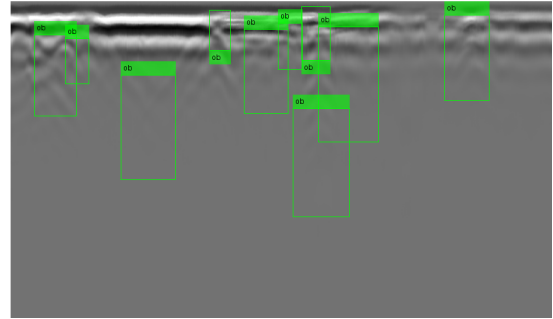
In Figure 5, we compare the detection results on a real GPR image yielded by our two scenarios compared to the COD based on HOG and Haar-like features. Here, results are shown without any post-processing technique. As we can observe, the Faster-RCNN can provide better performance compared to COD method which yielded unstable object bounding boxes and more false alarms. Importantly, adding more simulated data could provide detection results with higher accuracy and confidence, which validates our intention and confirms the effectiveness of the proposed scheme.

5. CONCLUSION AND FURTHER WORK

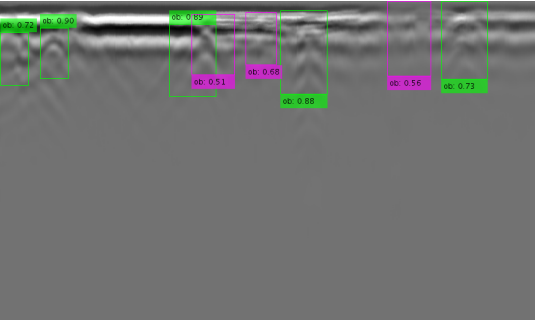
We have applied the well-known Faster-RCNN framework to the detection of buried objects from GPR B-scan data. By combining both simulated and real collected radargrams



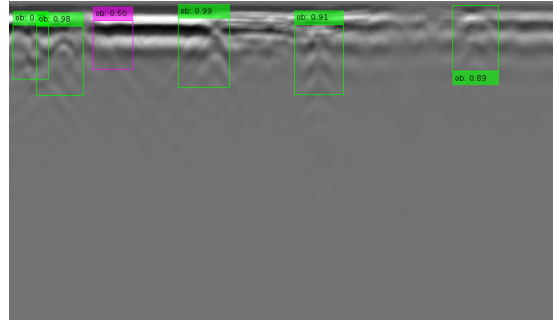
(a) Cascade object detector - HOG



(b) Cascade object detector - Haar-like



(c) Faster-RCNN scenario 2



(d) Faster-RCNN scenario 3 (best)

Fig. 5. Preliminary detection results on real data.

for training, the proposed technique can provide good performance on tested real data and considerably outperforms detectors using classical features such as HOG and Haar-like. Therefore, it becomes promising to deal with GPR data with limited training samples.

Although the effectiveness of the proposed scheme has been qualitatively observed, our on-going work now focuses on quantitative evaluation for a better validation. Moreover, further work related to the detection of hyperbola coordinates (apex and prong) is necessary for a fine and accurate localization of detected objects.

6. REFERENCES

- [1] C. G. Windsor, L. Capineri, and P. Falorni, "A data pair-labeled generalized Hough transform for radar location of buried objects," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 124–127, 2014.
- [2] W. Li, X. Cui, L. Guo, J. Chen, X. Chen, and X. Cao, "Tree root automatic recognition in ground penetrating radar profiles based on randomized Hough transform," *Remote Sensing*, vol. 8, no. 5, p. 430, 2016.
- [3] F. Sagnard and J.-P. Tarel, "Template-matching based detection of hyperbolas in ground-penetrating radargrams for buried utilities," *J. Geophys. Eng.*, vol. 13, no. 4, p. 491, 2016.
- [4] G. Terrasse, J.-M. Nicolas, E. Trouvé, and É. Drouet, "Automatic localization of gas pipes from gpr imagery," in *EUSIPCO*, pp. 2395–2399, IEEE, 2016.
- [5] P. A. Torrione, K. D. Morton, R. Sakaguchi, and L. M. Collins, "Histograms of oriented gradients for landmine detection in ground-penetrating radar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 3, pp. 1539–1550, 2014.
- [6] C. Maas and J. Schmalzl, "Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar," *Computers & Geosciences*, vol. 58, pp. 116–125, 2013.
- [7] L. E. Besaw and P. J. Stimac, "Deep convolutional neural networks for classifying GPR B-scans," in *SPIE Defense+ Security*, pp. 945413–945413, International Society for Optics and Photonics, 2015.
- [8] S. Lameri, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro, "Landmine detection from GPR data using convolutional neural networks," in *EUSIPCO*, pp. 508–512, IEEE, 2017.
- [9] D. Reichman, L. M. Collins, and J. M. Malof, "Some good practices for applying convolutional neural networks to buried threat detection in Ground Penetrating Radar," in *IWAGPR*, pp. 1–5, IEEE, 2017.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, pp. 91–99, 2015.
- [11] C. Warren, A. Giannopoulos, and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Computer Physics Comm.*, vol. 209, pp. 163–170, 2016.
- [12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, vol. 1, pp. I–I, IEEE, 2001.