Enhanced Underground Object Detection with Conditional Adversarial Networks

RICE Will, OMWENGA Maxwell, WU Dalei, LIANG Yu

College of Computer Science and Engineering, University of Tennessee at Chattanooga
Chattanooga, Tennessee, USA
Email: {jrg741, dgh179}@mocs.utc.edu; {dalei-wu, yu-liang}@utc.edu

Keywords: Ground Penetrating Radar, Generative Adversarial Networks, Underground Object Detection, Classification

Abstract - To augment training data for machine learning models in Ground Penetrating Radar (GPR) data analysis, this paper focuses on the generation of realistic GPR data using Generative Adversarial Networks (GAN). An innovative GAN architecture is proposed for generating GPR B-scans, which is, to the author's knowledge, the first successful application of GAN to GPR data. As one of the major contributions, a novel loss function is formulated by merging frequency domain features with time domain features. To test the efficacy of generated B- scans, a real-time object classifier is proposed to measure the performance gain derived from augmented B-Scan images. The numerical experiment illustrated that, based on the augmented training data, the proposed GAN architecture demonstrated a significant increase (from 82% to 98%) in the accuracy of the object classifier.

1. Introduction

Model performance in machine learning is heavily dependent upon the availability of training data. Furthermore, in situ applications of machine learning require data to be indicative of the real-world environment the applications aims to infer in. For buried object detection, this would be a B-scan image that bears a close resemblance to B-scans collected in the field. However, these data are not widely available, and when available, the number of samples is few. In situations like this, data augmentation can produce many samples to enhance the classification of buried objects [1]. In the world of GPR, an open source software named gprMax is used to simulate the presence of underground objects [2]. This software is based on Finite Difference Time Domain (FDTD) [3], a numerical method to solve Maxwell's equations that govern waves propagation within a specific medium. The problem with this type of simulated data is that it bears little resemblance to a B-scan that would be obtained in the real world. Furthermore, due to the complexity of FDTD, the time to complete a single simulation can take several hours. Thus, making the synthesis of a large set of images for training data, virtually impossible.

To solve this problem, we propose a novel generative model architecture to synthesize realistic B-scans in real time. In addition, to benchmark the generative model, a classifier is produced that is capable of running in real time on an edge computing server. The objectives of this study are three-fold. The first is to establish a generative architecture for the generation of pseudo realistic B-scans. The second is to develop a real-time classifier, capable of being deployed on an edge computing sever. The final is to incorporate frequency information into time-domain architectures.

In this work, GANs will be investigated for GPR data augmentation for object identification. Both simulated and real GPR data will be considered as inputs to the generator of a GAN to generate realistic GPR data. Based on the feature analysis of GPR data, a novel objective function and the architecture of GAN are proposed. An algorithm for GAN training with different types of training data is developed. The impact of GAN-synthesized data on the performance of GPR image classification is evaluated. To the best knowledge of the authors, very few works have been done on studying GANs for GPR data analysis in a united frame-work combining data augmentation and data classification. A detailed diagram of the overall system structure is depicted in Figure 1.

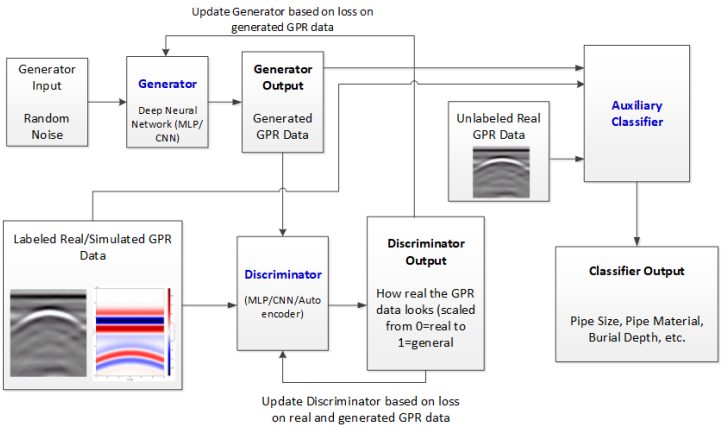

Figure 1. Proposed GAN architecture.

2. Background

GPR is one of the most widely used non-destructive techniques for subsurface imaging and detection of underground objects, such as landmines, utilities, and archaeological artifacts.

In the GPR scanning process, an electromagnetic wave is propagated into the target subsurface medium through a transmitting antenna, and upon reflection of the underground object, returned to a receiving antenna. This process is carried out across the above-ground surface for multiple passes.

2.1. B-scan Feature Processing

B-scans, along with other features, are commonly analyzed to detect or identify subsurface objects [1]. As an alternative to visual examination of B-scans by GPR technicians, machine learning techniques have been applied to analyze B-scans for object detection [1], [4]–[7]. By combining Hilbert transform and classic artificial neural network (ANN), the work in [4] used amplitude and time from GPR A-Scan to detect the shape, material and depth of a buried object. Extracting a signal envelope, peak detection of envelope and depth of buried empty tube from A-Scan through analytic signal technique [8]. Gilmore et.al [6] extracted features using the Hu's seven invariant moments algorithm, and latter passed them through an ANN classifier [5] to detect targets, however many false negatives were observed. While these techniques have been modestly effective, their performance is limited by the insufficient amount of real-world labeled GPR datasets for training the corresponding models or classifiers. To deal with the scarcity of GPR data, simulation-based methods have been proposed to increase the availability of training data, but these methods fail to represent the full spectrum of features found in real GPR data. Therefore, classifiers trained on simulated B-scan images tend to perform poorly on real world B-scan images. A successful remedy is the combination of simulated and collected real-world GPR data in training [1].

2.2. Generative Adversarial Networks and Applications

A GAN is a form of generative model, in which two separate models are entangled in a zero-sum game. The generative portion of the network is denoted as the generator (G). The goal of the generator is to synthesize the most realistic posterior distribution. In opposition of the generator, the discriminator (D), decides if the posterior distribution is legitimate, or a counterfeit. This process is carried out in tandem during training and can lead to instability [9]. The input to the generator is a Gaussian noise vector N(0,1). A transformation is applied to this vector, thus producing a posterior with equal dimensionality as the target distribution. Furthermore, this process is carried out by interpolation of the input vector through one or more deconvolution operations. The discriminator decomposes this output distribution into a binary probability through a sigmoid activation. The basics of this min-max game have changed very little since first proposal. Generative Adversarial Networks have received wide attention in the machine learning field for their potential to learn high-dimensional, complex real data distribution [9], [10]. Specifically, they do not rely on any assumptions about the distribution and can generate real-like samples from latent space in a simple manner. This powerful property leads GANs to be used in many generative tasks to replicate the real-world rich content such as images, videos, speech, written language, and music [10].

There has been some work done on employing GANs for data augmentation in image classification using deep learning [10]. Furthermore, GAN can also be interpreted to measure the discrepancy between the generated data distribution and the real data distribution and then learn to reduce it. The discriminator is used to implicitly measure the discrepancy. Despite the advantage and theoretical support of GAN, many shortcomings have been found due to the practical issues and inability to implement the assumption in theory including the infinite capacity of the discriminator. There have been many attempts to solve these issues by changing the objective function, the architecture, etc. Moreover, the most recent additions to the adversarial framework have improved on many weak points in the original architecture. Wasserstein Loss has been used in GAN models to improve the stability of the adversarial game [11]. Moreover, this architecture can be further improved with the use of a gradient penalty term [12].

2.3. Evaluation of Generative Models

Many generative architectures use Mean Opinion Score (MOS) or other qualitative metrics for model evaluation [13]. This type of evaluation is readily available via Amazon Turkers or similar service that allows the general public to give an opinion. In our case, qualitative evaluation is unrealistic due to the requirement of domain expertise to detect the realistic nature of each synthesized GPR signal. Moreover, we would like to stray away from qualitative evaluation and use a more quantitative method. In this work, we validate the quality of generated output by an improvement factor in the recognizant ability of our object identification model. It is important to note the parallels between this technique and the commonly used Inception Score [14]. However, with the lack of widespread availability of benchmark classifiers applied to this domain, we use a different method.

3. Methodology

To acquire training data for the GAN model, we use gprMax, an open source software to simulate electromagnetic wave propagation. It solves Maxwell's equations in 3D using the Finite Difference Time Domain (FDTD) method [2]. We generate cylinders with diverse dielectric properties in range of substrate mixtures. For purposes of sample diversity, we focus on a range of cylinder diameters in Peplinksi soil [15], with a range of sand to clay ratio for each image. To provide additional randomness, we apply a seed value that is randomly selected and applied to each iteration of training data generation. Therefore, each image produced by gprMax is unique. A total of 150 A-scan traces comprise the B-scan of a single simulation, as shown in Figure 2.

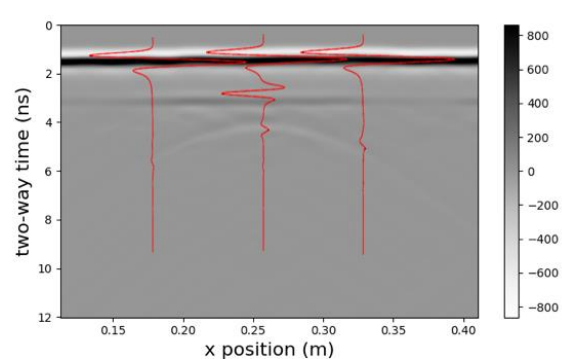

Figure 2. A B-scan includes 150 A-scan traces.

The architecture proposed is a deep convolutional structure. Previous work suggested that a convolution with the filter with dimension of (5,5) is superior to other options for the modeling of GPR data [16]. Therefore, we set the kernel size of all convolutions to 5 by 5. The generator is conditioned to upsample a noise vector into a class from a supervised label. This is accomplished by introducing a label embedding vector and concatenating it with the posterior of the generator [17]. From this output, we calculate Wasserstein loss with gradient penalty [12] against the true image. The discriminator is used to determine the validity of the generated output by directly comparing the two images. We calculate the Wasserstein distance between both real and fake images. In addition, the discriminator is trained to produce a predicted label for the generated image. For this output, we calculate categorical cross entropy between the predicted label and the true label. To improve the overall model quality, we apply a frequency domain loss function to G(z).

The generator takes the inputs of a random noise vector and a label for the desired class. The label is then passed through an embedding layer which allows for multiplication with the noise tensor. This is the vital step for the introduction of class conditioning and leaves us with a single input for the remaining layers. Next, the combined input is passed through a dense layer which gives us the dimensionality to be able to reshape the tensor into the 3-Dimensional shape of an image. From this point, we begin the upsampling process. We derive the upsampling method from [18], which indicates that many upsampling layers are favorable. In addition, we pass the upsampled vector though a convolution layer. This allows us to retain only the important information that we upsampled. Each time the input passes through an upsampling layer it doubles in size. The output is then activated with ReLU [19] and then passed through a batch normalization layer for regularization. We continue this process until the generator input is the same size as our target image. Finally, a Tanh [20] activation is applied to restrict the output to the range (-1, 1). The important part of the generator is that we want to learn the transformation of a noise vector into an image. We use a Gaussian noise vector because it contains the least amount of prior knowledge [21]. Therefore, the primary learning objective is not what the generator learns from the noise vector, but how we can exploit the functional approximation property of neural networks to transform the noise vector into an image.

The discriminator accepts a tensor in the shape of a real image (256, 256, 1). During training, it receives both real and fake images and directly compares the two. This feedback is used to condition the generator to make better images. We use LeakyReLU [22] activation to reduce mode collapse, because the gradient after this activation is never 0. The downsampling pattern is a reversed version of the upsampling pattern in the generator. This adds additional balance to the training process which produces additional stability. Furthermore, we flatten the tensor before it enters the final dense layer. This can be thought of as a summary of the information learned in the previous layers. Note, there is not a non-linearity applied to the output of the final layer. This is used for direct comparison with the real image.

We train a separate auxiliary classifier to predict the object in the image. This is a basic classifier that uses cross entropy to create a separation boundary between classes. The architecture consists of two convolution layers that lead into a fully connected layer. The output of the final fully connected layer is activated with SoftMax to generate a categorical probability distribution. The loss function is traditional categorical cross entropy. We apply this loss to both the Time B-scan and Frequency B-scan to maximize the probability of a correct class prediction. Table 1 depicts the basic classifier architecture. The basic classifier has two convolution layers, both activated with Leaky ReLU [22]. We use this opposed to traditional ReLU to mimic the architecture of the discriminator. In the initial tests we sought to use the discriminator as the classifier. However, this leads to extreme over fitting in the discriminator and poor performance for the classification task. Moreover, this also had a negative effect in the adversarial game, with the generator being able to constantly fool the discriminator. An important note in using a separate classifier is that this simple architecture can be a stand in for more complex object detection models such as Faster-RCNN [23] or Mask-RCNN [24]. This was an additional reason for not using the discriminator as the object detection model. To enable the use of the Time B-scan and Frequency B-scan, the architecture has a slight modification as depicted in Table 2. The additional fully connected layer allows us to calculate a separate loss for the Frequency B-scan which is useful in training.

Table 1. Single classifier architecture.

Operation	Output Shape
Input B-scan	(n, 256, 256, 1)
Conv2D(2, 1, strides=2)	(n, 128, 128, 2)
LeakyReLU(alpha=0.3)	(n, 128, 128, 2)
Conv2D(4, 1, strides=2)	(n, 64, 64, 4)
LeakyReLU(alpha=0.3)	(n, 64, 64, 4)
Flatten	(n, 16384)
Dense	(n, 3)

Table 2. Combined classifier architecture

Operation	Output Shape
Input Time B-scan	(n, 256, 256, 1)
Input Frequency B-scan	(n, 256, 256, 1)
SingleClassifier(Time B-scan)	(n, 64, 64, 4)
SingleClassifier(Frequency B-scan)	(n, 64, 64, 4)
Multiply	(n, 64, 64, 4)
Flatten	(n, 16384)
Dense	(n, 3)

The auxiliary classifier is trained in three scenarios. The first, data containing only images from gprMax are adopted. We use this performance as a baseline to compare our other experiments. The second, we use the full gprMax generated dataset with additional GAN generated images. Finally, we add a concurrent frequency domain optimization function to the generator, then train the object detection model to identify cylinder material from the b-scan with the assistance of information from the frequency representation.

4. Experimental Results and Discussion

In this section, we study the results of supervised GAN Experiments. These are the set of experiments that contain class conditioning. This is only possible with the use of GprMax that allows us to simulate the material of the underground object and retain a definitive label. This is necessary in the classification task and also a major drawback of collected B-scans. In the field, B-scans that are collected do not have a ground truth class label due to the subjective nature of real B-scan evaluation. In this experiment, we can generate realistic type data that does have a definitive class label. Therefore, we are able to map a set of image features to the label.

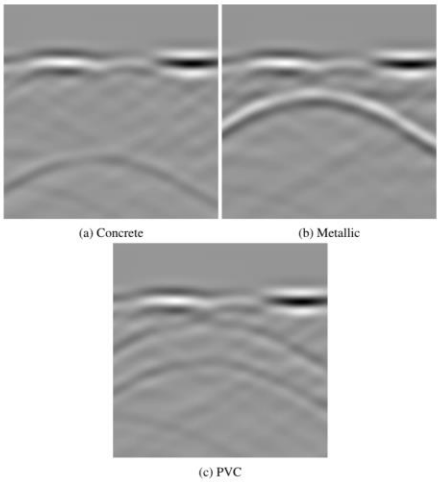

Figure 3. B-scan examples of the three different target classes.

Figure 3 shows B-scan examples of the different target classes. The GAN model was able to learn distinct features of each class and then generate images of this type when given a label. An important note on continuation of this work is that ideally one would want to be able to combine the unsupervised with the supervised to generate a real B-scan with a known class. This is theoretically possible; however, it is beyond the scope of this work. The following sections are the performance of the proposed classifier in each test scenario. The main objective is to demonstrate improvement in two aspects. We would like to see performance improvement with data augmentation via GAN generated images and further improvement when combining the time and frequency B-scans.

4.1. Time Domain

Table 3 presents the results of the baseline performance. It is important to point out that these results are still in the upper half in regard to the performance metrics. However, as it will be demonstrated, there is still room for improvement. Class "PVC" is by far the worst performing material class. This is due to the lack of reflectivity in PVC cylinders. In Figure 3, it can be seen that the B-scan of "PVC" is visually, the least prominent hyperbola followed closely by that of "Concrete". From the results, this visibility difference translates to the classifier performance. "Metallic", being the most visually prominent, is easily identified by the detection model. The lower portion of Table 3 shows the performance achieved by training the classifier with augmented data. Overall, there is a performance increase when adding augmented training data. Most importantly, this is seen in the weak areas of the classifier. In the accuracy of "PVC" there is significant improvement that closes the gap between "PVC" and "Metallic". This means that when more samples are present in the training data an overall increase in classifier performance will be realized. The next section discusses a classifier only trained on the frequency domain information of B-scans. This experiment was to determine if the frequency representation leads to better performance in a particular class.

Table 3. Baseline object detection performance.

Before Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	1.00	0.54	1.00	0.70	20
Metallic	0.81	1.00	0.82	0.90	11
PVC	0.50	0.67	0.57	0.62	14
All	0.77	0.74	0.80	0.75	45
After Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	1.00	0.88	1.00	0.93	14
Metallic	0.91	1.00	0.91	0.95	11
PVC	0.90	0.95	0.90	0.92	20
All	0.94	0.94	0.94	0.93	45

4.2. Frequency Domain

Table 4 contains the numerical performance results. The classification of class "PVC" outperformed that of metallic in accuracy when using a frequency B-scan. The significance in this is that if frequency information is given to the classifier that the weakest class in the baseline is able to be detected at a better rate than the strongest performing baseline class. Notice that precision in the frequency domain is high in all of the classes. Recall is an additional area in which class "PVC" performs well. Although, performance is not quite as good as the time domain classifier trained with augmented data. Next, let us look at how augmentation can improve performance in the frequency domain. The bottom half of Table 4 contains the metrics after augmentation. Overall, there is improvement in all metrics.

Table 4. Object detection performance with frequency domain information.

Before Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	0.21	0.80	0.21	0.33	19
Metallic	0.18	1.00	0.62	0.77	16
PVC	0.20	0.75	0.90	0.51	10
All	0.20	0.86	0.58	0.54	45
After Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	0.64	0.83	1.00	0.90	19
Metallic	0.36	1.00	0.88	0.93	16
PVC	0.60	1.00	0.60	0.67	10
All	0.53	0.94	0.83	0.83	45

4.3. Combined

In the combined approach, we are using both time domain and frequency domain features of B-scans. This essentially is doubling the number of features used for classification. As reviewed in previous sections, frequency representations allowed improvement for the weak areas of object material classification. A combined approach will yield an improved classifier for all materials.

Table 5. Object detection performance with combined domain information.

Before Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	1.00	0.54	1.00	0.70	18
Metallic	0.91	1.00	0.93	0.96	14
PVC	0.55	1.00	0.69	0.62	13
All	0.82	0.85	0.87	0.76	45
After Augmentation					
Class	Accuracy	Precision	Recall	F1	N
Concrete	1.00	0.88	1.00	0.93	14
Metallic	1.00	1.00	1.00	1.00	11
PVC	0.95	1.00	0.90	0.95	20
All	0.98	0.96	0.97	0.96	45

Table 5 depicts the classification scores achieved before the use of augmented data. Compared to the baseline, this approach realizes a significant increase in all evaluation metrics. Notice that class "PVC" is still the class having the worst performance in accuracy. However, an improvement from the baseline can still be observed. This indicates that using a combined approach did improve the results of a weak class in accuracy. This is also true for the other metrics in relation to class "PVC". The class "Concrete" did not see an improvement from the baseline when adding features from the frequency domain. This is unusual due to the increased performance in all metrics from the frequency domain experiments. However, it is important to note that class "Concrete" already achieved max values in accuracy and precision in the baseline test. Therefore, the improvement did not occur in precision only. The two other classes, "Metallic" and "PVC", saw performance improvement in every metric with the combined approach. Thus, a combined approach is superior to the approach using only time domain or frequency domain features.

5. Conclusion

This paper explored generative models for GPR. With labeled training data, a conditional generative architecture were applied for GPR data augmentation. Furthermore, it was shown that a real-time classifier can be trained to detect the material of underground objects, and that this model can be improved with the incorporation of frequency domain features in classification. Moreover, with the addition of GAN synthesized data, we can train a classifier that detects objects with very high classification scores.

6. Acknowledgement

This work was supported by the National Science Foundation under grant number 1647175.

7. References

[1] M.-T. Pham and S. Lefevre, "Buried object detection from b-scan ground penetrating radar data using faster-rcnn," CoRR, vol. abs/1803.08414, 2018.

[2] C. Warren, A. Giannopoulos, and I. Giannakis, "gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," Computer Physics Communications, vol. 209, pp. 163 – 170, 2016.

[3] R. M. Joseph and A. Taflove, "Fdtd maxwell's equations models for nonlinear electrodynamics and optics," IEEE Transactions on Antennas and Propagation, vol. 45, no. 3, pp. 364–374, 3 1997.

[4] Y. Zhang, D. Huston, and T. Xia, "Underground object characterization based on neural networks for ground penetrating radar data," in Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2016, vol. 9804, 2016.

[5] N. P. Singh and M. J. Nene, "Buried object detection and analysis of gpr images: Using neural network and curve fitting," in ICMiCR 2013.

[6] C. Gilmore, et al., "Gpr target detection using a neural network classifier of image moments as invariant features," in ANTEM/URSI 2004.

[7] J. S. Kobashigawa, et al., "Classification of buried targets using ground penetrating radar: Comparison between genetic programming and neural networks," IEEE Antennas and Wireless Propagation Letters, vol. 10, pp. 971–974, 2011.

[8] R. Ghozzi, et al., "Extraction of geometric parameters of underground tube using gpr," in ICCAD 2017.

[9] I. J. Goodfellow, et al., "Generative Adversarial Networks," ArXiv e-prints, Jun. 2014.

[10] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial nets and its variants work: An overview of GAN," CoRR, vol. abs/1711.05914, 2017.

[11] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in ICML 2017.

[12] I. Gulrajani, et al., "Improved training of wasserstein gans," in NIPS 2017.

[13] Q. Huynh-Thu, et al., "Study of rating scales for subjective quality assessment of high-definition video," IEEE Transactions on Broadcasting, vol. 57, no. 1 2011.

[14] T. Salimans, et al., "Improved Techniques for Training GANs," arXiv e-prints, p. arXiv:1606.03498, 2016.

[15] N. R. Peplinski, et al., "Dielectric properties of soils in the 0.3-1.3-ghz range," IEEE TGRS, vol. 33, no. 3, 1995.

[16] M. Almaimani, et al., "Classifying gpr images using convolutional neural networks." EAI, 9 2018.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," CoRR, vol. abs/1411.1784, 2014.

[18] E. L. Denton, et al., "Deep generative image models using a laplacian pyramid of adversarial networks," CoRR, 2015.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in ICML 2010.

[20] A.-M. Wazwaz, "The tanh and the sine-cosine methods for compact and noncompact solutions of the nonlinear klein-gordon equation," Appl. Math. Comput. 2005.

[21] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning MITPress, 2016.

[22] B. Xu, et al., "Empirical evaluation of rectified activations in convolutional network," CoRR, 2015.

[23] S. Ren, et al., "Faster R-CNN: towards real-time object detection with region proposal networks," CoRR, 2015.

[24] K. He, et al., "Mask R-CNN," CoRR, 2017.