IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 20, 2021, accepted February 27, 2021, date of publication March 8, 2021, date of current version March 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3064205

A Gans-Based Deep Learning Framework for Automatic Subsurface Object Recognition From Ground Penetrating Radar Data

XIN ZHANG[1], LIANGXIU HAN[1], MARK ROBINSON[2], AND ANTHONY GALLAGHER[3]

[1]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, U.K.
[2]School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.
[3]Railview Ltd., Northumberland NE65 8QY, U.K.

Corresponding author: Liangxiu Han (l.han@mmu.ac.uk)

This work was supported by the Innovate UK funded project [Infrastructure Monitoring System (INFRAMONIT)] under Grant 104245.

ABSTRACT Ground penetrating radar (GPR) is a well-known useful tool for subsurface exploration. GPR data can be recorded at a relatively high speed in a continuous way with hyperbolas being artifacts and evidence of disturbances in the soil. Automatic and accurate detection and interpretation of hyperbolas in GPR remains an open challenge. Recently deep learning techniques have achieved remarkable success in image recognition tasks and this has potential for interpretation of GPR data. However, training reliable deep learning models requires massive labeled data, which is challenging. To address the challenges, this work proposes a Generative Adversarial Nets (GANs)-based deep learning framework, which generates new training data to address the scarcity of GPR data, automatically learns features and detects subsurface objects (via hyperbola) through an end-to-end solution. We have evaluated our proposed approach using real GPR B-scan images from rail infrastructure monitoring applications and compared this with the state-of-the-art methods for object detection (i.e. Faster-RCNN, Cascade R-CNN, SSD and YOLO V2). The proposed approach outperforms the existing methods with high accuracy of 97% being the mean Average Precision (mAP). Moreover, the proposed approach also demonstrates the good generalizability through cross-validation on independent datasets.

INDEX TERMS Generative adversarial nets (GANs), deep learning, ground penetrating radar (GPR).

I. INTRODUCTION

Ground penetrating radar (GPR) has been considered as a useful tool for underground imaging and detection of buried objects, such as mines, pipes, cultural relics in different domains (archaeology, military, etc.) [1]. During the GPR scanning process, the electromagnetic wave is transmitted to the target underground through the transmitting antenna and returns to the receiving antenna after being reflected by the underground object. As a result, the hyperbolic image features can be extracted as a 2D image from the buried objects by merging all the signal data. Analyzing the shape and location of the hyperbolas is essential for locating, determining the diameter and detecting the anomalies of subsurface objects.

However, the automatic detection of hyperbolas in a GPR dataset is a challenging task. Different methods have been proposed to automatically detect buried objects using GPR B-scan images [2]. Several hand-crafted features including Hough transform (HT), Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) are generally combined with machine learning classification algorithms such as Support Vector Machines and Random forest to detect hyperbolas [3], [4]. Despite encouraging work, most of these methods could only deal with a small amount of data but with the complicated process for extracting hand-crafted features, which is inefficient.

Recently, deep Learning (DL) has achieved remarkable success in image recognition tasks including image classification, segmentation and object detection [5]. Efforts have also been made in GPR image processing. The work in [6] used a deep convolutional neutral network (DCNN) to classify B-scan profiles into threat and non-threat classes.

The associate editor coordinating the review of this manuscript and approving it for publication was Sara Dadras.

The results showed that DCNN could extract meaningful features and signatures complementing existing GPR feature extraction and classification techniques. In [7], landmine detection based on GPR data using DCNN has showed promising results compared with traditional computer vision methods. In [8], the authors reviewed and applied several DCNN models to the detection of buried objects from GPR data.

The most challenging aspects preventing the application of deep learning models is the limited amount of training datasets. In order to solve the shortage of GPR data, simulation-based methods were proposed to increase the availability of training data. GprMax, an open-source software was widely to use for simulating GPR data [1], [9]. However, using GprMax to generate simulated GPR data is time consuming, even with CUDA support, simulation can take many hours on advanced GPU cards.

Generative Adversarial Nets (GANs) have recently attracted a lot of attention in the machine learning area for generating new synthetic data with the same statistics as the training set [10]. It has a capability to learn and mimic any distribution of data and has been widely applied in image generation, video generation, etc [11], [12].

In this work, we have proposed a GAN-based deep learning framework for accurately automatic hyperbola detection through an end-to-end solution in order to address the challenges: 1) the scarcity of GPR training data; 2) efficiently automated detection of hyperbola. The contributions of this article include:

- Application and optimization of Generative Adversarial Networks to generate GPR data to address the scarcity of GPR data.
- Proposal for a single-stage object-detector with deep learning for automatically identifying hyperbola, enabling fast inference and easy training.

The remainder of this article is organized as follows: Section II presents the related work; Section IIIdetails the proposed method; Section IVdescribes experimental evaluation; Section V concludes the work.

II. RELATED WORK

A. TRADITIONAL HAND-CRAFTED FEATURES BASED DETECTION OF BURIED OBJECTS METHODS

Different handcrafted feature-based methods have been proposed for automatic detection of buried objects using GPR B-scan images. The Hough transform (HT) was first and most widely used for deformed shape fitting. In [3], [4], the generalized and the randomized HT features were extracted from B-scan images to determine the hyperbola parameters recorded within the Hough accumulator space. In [13], Scale Invariant Feature Transform (SIFT) was employed for feature extraction to localize the hyperbola regions.

In [14], [15], the template matching and dictionary-based techniques were used to match hyperbola features. The correlation scores between each images of template were used to evaluate the model. In [16], the Haar-like features was used by learning Viola-Jones Algorithm for automatically detecting reflection hyperbolas in raw GPR data. The Histogram of oriented gradients (HOG) feature were used in supervised pattern recognition approach for hyperbola detection [17].

Despite the existing efforts, the limitations of these traditional approaches persist, namely 1) the training datasets are small; 2) the selection process of handcraft features for each given task is complicated and inefficient; These result in model inaccuracy for automated hyperbola detection.

B. GENERATIVE ADVERSARIAL NETS

Generative Adversarial Nets (GAN) have achieved remarkable success and can be used to generate a large amount of training data for deep learning model training. GANs were firstly used to generate new samples for the MNIST handwritten digit dataset [10]. In [18], GANs was used to generate plausible realistic photographs of human faces. In [19], the work proposed GANs model for generating new 3D objects in ten popular object categories, such as guns, chairs, doors, balls and tables. In [20], the unsupervised GANs approach was used to impute new ground penetrating radar data based on limited and class imbalance labeled data which improved the detection model significantly. Thomas Truong *et al.* used GANs to generate radar signal which was indistinguishable from the training data by human observers [21].

However, training GANs is not easy. GANs models may suffer from the several problems including mode collapse, diminished gradient, non-convergence, etc [22]. There are a plethora of methods such as Alternative Loss Functions like Wasserstein GAN [23], Gradient Penalty [24], Spectral Normalization and Self Attention Mechanism [25] were used for addressing previously mentioned issues and making GANs more stable and powerful.

C. DEEP LEARNING-BASED OBJECT DETECTION METHODS

Different from handcrafted feature-based methods, the DCNN methods skip the traditional feature selection and parameters tuning steps. Instead, they learn the feature representations of buried objects directly by convolutional operation from GPR B-scan images. Several researchers have shown remarkable performance with hyperbola recognition and detection based on deep learning methods [6], [8], [26], [27].

At present, the research of object detection based on DCNN can be divided into two categories: **multi-stage detection** and **single stage detection**. The multi-stage methods can be broadly divided into two stages that start with the region search and then determine the classification. The Region Proposal Network (RPN) is the most well-known region search method and have proven to be very efficient till now. It aims to propose multiple objects that are identifiable within a particular image. Faster Region-based Convolutional Network (Faster R-CNN) from Facebook [28] and Cascade R-CNN from Multimedia Laboratory, CUHK [29] are the two most famous and widely used multi-stage detection models. The single stage detection model is also named as one-time prediction model. It predicts bounding boxes and class probabilities using a single network in a single evaluation directly. This makes real-time predictions possible. You Only Look Once (YOLO) series is the first single stage detection model [30]–[32], which was designed for rapid inference. For the purpose of improve its accuracy, Single-Shot Detector (SSD) [33] and RetinaNet [34] are designed by optimizing YOLO architecture and performed the excellent accuracy.

III. THE PROPOSED METHOD

To address aforementioned challenges, in this work, we have proposed a GAN-based deep learning framework, which generates new training data to addresses the scarcity of GPR data and automatically learn features and detect hyperbolas through an end-to-end solution.

The overview of the proposed framework is shown in Figure 1. The framework consists of two main parts: 1) Data generation through the proposed GANs; 2) Object detection based on deep learning.

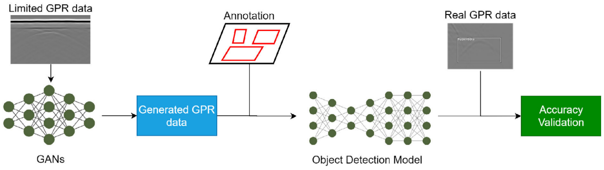


FIGURE 1. **The overview of automatic detection of hyperbolas.**

A. DATA GENERATION WITH THE PROPOSED GANs

In order to generate new data for the purpose of model training, we have applied a GANs for generating GPR B-scan images. The architecture of the proposed GANs is shown in Figure 2.

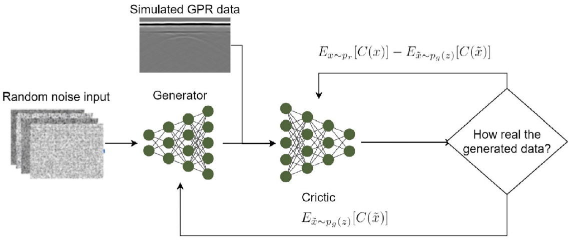

Simulated GPR data
$E_{x\sim p_r}[C(x)] - E_{\tilde{x}\sim p_g(z)}[C(\tilde{x})]$
Random noise input
Generator
How real the generated data?
Crictic
$E_{\tilde{x}\sim p_g(z)}[C(\tilde{x})]$

FIGURE 2. **The architecture of the proposed GANs.**

The concept of GANs is that two models are trained at the same time: a generator (G) and a critic (C) model. The generator tries to make new data similar to the data in our dataset, and the critic tries to classify real images from the ones produced by the generator.

Most GANs models suffer from the following problems:

- The model parameters are unstable and never converge;
- The generator collapses during the training stage;
- The generator gradient vanishes and learns nothing.
- The traditional cost functions may not converge using gradient descent during the adversarial training.

In our work, the generator and critic models are both built by convolutional networks. The input of G is a random noise vector and the output is a fake GPR B scan image. The input of the C is the faked image from G and the simulated GPR data (real image). It is used to evaluate how the generated data truly estimates the probability that the sample comes from the training data rather than G. The Wasserstein distance was selected to evaluate similarity, which was designed to measure the difference between the data distributions of real and generated images [35]. It can be used to smooth the gradient that improves the stability of the adversarial training. The training procedure for G is to maximize the probability of C making a mistake. This method was considered as a minimax two-player game. The minimax objective is used to train both G and C and can be formulated as:

$$\min_{G}\max_{C\in w} V(C, G) = E_{x\sim p_r}[C(x)] - E_{\tilde{x}\sim p_{g(z)}}[C(\tilde{x})] \quad (1)$$

The minimax objective encourages G to fit p_r so as to fool C with its generated samples $C(x)$. Where $E_{x\sim p_r}[C(x)]$ means the expected distribution of function when input is x. $x \sim p_r$ means sample x that has probability distribution p_r. $p_g(z)$ is model distribution implicitly defined by $\tilde{x} = G(z)$, $z \sim p(z)$. In this work, the Wasserstein distance w was used to evaluate the critic model, where $max_{c\in w}$ means the maximum of w in the Wasserstein distance.

The w is a set of $1 - Lipschitz$ functions that requires the weights of the function to be within a range controlled by hyper parameter u.

$$w \leftarrow (w, -\mu, \mu) \quad (2)$$

However, in the training stage, the range μ is not easy to set. The gradient disappears when μ decreases and the gradient explodes when μ increases a little. In fact, we encourage the norm of the gradient to reach 1 instead of just staying below 1. To achieve this aim, we added a gradient penalty in the 1-Lipschitz function. The final loss of the critic is defined by Equation 3 which includes the gradient penalty.

$$Loss = E_{x\sim p_r}[C(x)] - E_{\tilde{x}\sim p_{g(z)}}[C(\tilde{x})] + \lambda(\left\|\nabla_{\hat{x}} C(\hat{x})\right\|_2 - 1)^2 \quad (3)$$

where the $\lambda(\left\|\nabla_{\hat{x}} C(\hat{x})\right\|_2 - 1)^2$ is the gradient penalty term, $\widehat{x}$ means sample from x and $\tilde{x}$ with a uniformly t between 0 and 1.

$$\widehat{x} = (1 - t)x + t\widetilde{x} \quad (4)$$

The aim of the generator is to generate an image which takes the input of a random noise vector. Then several up-sampling operations are set to reshape the input vector to the defined size of the generated image (512 × 512 in this work). The design of generator involves a transposed convolutional block to up-sample the input. There are seven transposed convolutional blocks used in this work. Each block consists of a transposed convolutional layer, followed by a Batch Normalization and ReLU activation. In the first block, the kernel of transposed convolutional layer is

4 × 4 with stride as 1 that resizes the input vector to 4 × 4 with 4096 channels. Then the stride of transposed convolutional layer was set as 2 with a number of features divided by 2 at each block. A transpose convolution of size 4 × 4 and stride 2 followed by *Tanh* was set as the last block to resize the output as 1 × 512×512. The detailed architecture of the generator is shown in Table 1.

TABLE 1. The Architecture of the Generator Model.

Layer (type)	Output Shape	Training parameters
ConvTranspose2d	[1, 4096, 4, 4]	6,553,600
ReLU	[1, 4096, 4, 4]	0
BatchNorm2d	[1, 4096, 4, 4]	8,192
ConvTranspose2d	[1, 2048, 8, 8]	134,217,728
ReLU	[1, 2048, 8, 8]	0
BatchNorm2d	[1, 2048, 8, 8]	4,096
ConvTranspose2d	[1, 1024, 16, 16]	33,554,432
ReLU	[1, 1024, 16, 16]	0
BatchNorm2d	[1, 1024, 16, 16]	2,048
ConvTranspose2d	[1, 512, 32, 32]	8,388,608
ReLU	[1, 512, 32, 32]	0
BatchNorm2d	[1, 512, 32, 32]	1,024
ConvTranspose2d	[1, 256, 64, 64]	2,097,152
ReLU	[1, 256, 64, 64]	0
BatchNorm2d	[1, 256, 64, 64]	512
ConvTranspose2d	[1, 128, 128, 128]	524,288
ReLU	[1, 128, 128, 128]	0
BatchNorm2d	[1, 128, 128, 128]	256
ConvTranspose2d	[1, 64, 256, 256]	131,072
ReLU	[1, 64, 256, 256]	0
BatchNorm2d	[1, 64, 256, 256]	128
ConvTranspose2d	[1, 1, 512, 512]	1,024
Tanh	[1, 1, 512, 512]	0

The input of critic models is the fake images from generator and real images. During training, it compares two images and the output is used to adjust the generator to make images more real. As mentioned in the introduction, LeakyReLU activation was used to reduce mode collapse [36]. It includes a first 4 × convolutional blocks of stride 2 followed by 3 × 3 convolutional blocks of stride 1. Then we put seven 4 × 4 convolutional blocks of stride 2 with several features multiplied by 2 at each stage. The Lamda block was set as the last block as a summary of the information learned in the previous blocks. This is used for direct comparison with the real image. The detailed architecture of the critic is shown in Table 2.

TABLE 2. The Architecture of the Critic Model.

Layer (type)	Output Shape	Training parameters
Conv2d	[1, 64, 256, 256]	1,088
LeakyReLU	[1, 64, 256, 256]	0
Conv2d	[1, 64, 256, 256]	36,864
LeakyReLU	[1, 64, 256, 256]	0
BatchNorm2d	[1, 64, 256, 256]	128
Conv2d	[1, 128, 128, 128]	131,072
LeakyReLU	[1, 128, 128, 128]	0
BatchNorm2d	[1, 128, 128, 128]	256
Conv2d	[1, 256, 64, 64]	524,288
LeakyReLU	[1, 256, 64, 64]	0
BatchNorm2d	[1, 256, 64, 64]	512
Conv2d	[1, 512, 32, 32]	2,097,152
LeakyReLU	[1, 512, 32, 32]	0
BatchNorm2d	[1, 512, 32, 32]	1,024
Conv2d	[1, 1024, 16, 16]	8,388,608
LeakyReLU	[1, 1024, 16, 16]	0
BatchNorm2d	[1, 1024, 16, 16]	2,048
Conv2d	[1, 2048, 8, 8]	33,554,432
LeakyReLU	[1, 2048, 8, 8]	0
BatchNorm2d	[1, 2048, 8, 8]	4,096
Conv2d	[1, 4096, 4, 4]	134,217,728
LeakyReLU	[1, 4096, 4, 4]	0
BatchNorm2d	[1, 4096, 4, 4]	8,192
Conv2d	[1, 1, 1, 1]	65,536
Lambda	[1]	0

B. OBJECTION DETECTION

Inspired by the success of YOLO [32], We have proposed a one-stage detection model titled GPR-Detection. Compared with the multi-stage detection models, the single-stage model has a faster inference speed and is easier to train. The proposed GPR detector contains three main blocks: Backbone, Neck and Head. The proposed architecture is shown in Figure 3.

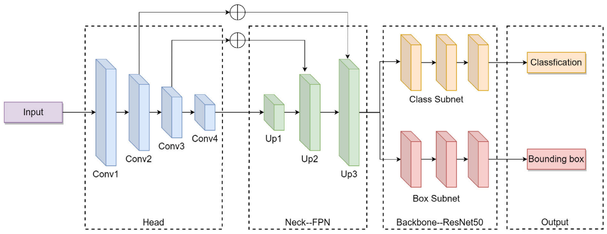


FIGURE 3. The architecture of the proposed object detection model (GPR-Detector).

1) BACKBONE

Backbone is the block to extract deep features from the origin image. Residual network is the most widely used backbone for many computer vision tasks. It adds skip connection between input and output after convolutional operation, which can ease the problem of vanishing or exploding gradients when the network becomes larger. In this work, the ResNet50 with 50 layers architecture was selected as the backbone (Figure 4). The output of Conv2, Conv3 and Conv4 was set as the input of Neck block.

2) NECK

The purpose of the neck block is to add extra layers between the backbone and the head blocks to extract different feature maps of various stages of the backbone. Feature Pyramid Networks (FPN) were selected as the neck block. It was introduced to use both bottom and upper layers in the feature map for detection, which provides a top-down pathway for construction of higher resolution features. FPN upsamples the previous output layers and adds it to the neighboring layer of the backbone (see Figure 3). It creates the feature maps at different scales for the head block. Then generate hierarchical

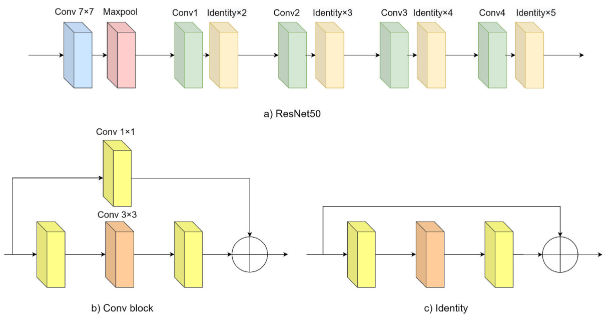


FIGURE 4. The architecture of ResNet50.

structure by using different spatial resolution feature maps from head block.

3) HEAD

The head block was used to get the classification and bounding boxes for each object by using classification and regression methods. A single output has four values describing the predicted bounding box (x, y, h, w) and the probability of hyperbola. In this work, we have applied the head network on each layer at different scales to get a better result.

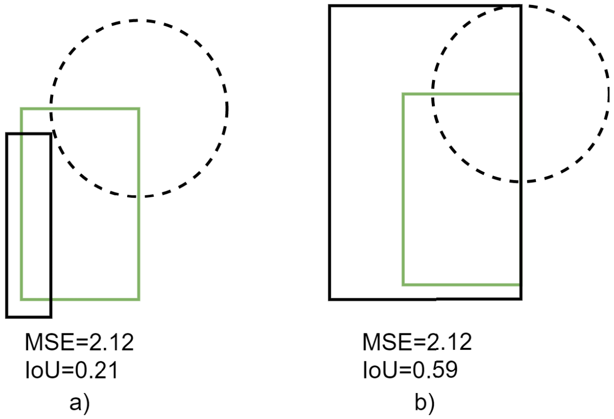


FIGURE 5. A example of the same MSE values with different IoU between ground truth and predicted bounding boxes.

The traditional loss function of the regression network is Mean Squared error which is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i) \tag{5}$$

However, in Equation 5, each point Y is treated as an independent variable, which doesn't consider the integrity of the object itself. It doesn't have a good correlation with the mean Average Precision (mAP). In Figure 5 where the green rectangle represents the ground truth and black rectangle represents the predicted bounding box. Due to the concept of MSE, any predicted bounding box where the second corner lies on a circle with a radius centered on the corresponding corner of ground truth will have the same MSE value. However, their Intersection over Union (IoU) values are different [37].

Intersection over Union has scale invariant and related with the ground truth directly. However, if IoU reaches zero between predicted and ground truth bounding boxes, the IoU cannot reflect if two shapes are in vicinity of each or extremely far from each other. To address this issue, a penalty term was added to minimize the normalized distance between central points of two bounding boxes and thus converges much faster than IoU [38]. The loss function is called the Distance-IoU Loss function and is defined as follows:

$$L_{DIoU} = 1 - \frac{\bigcap_{B}^{B^{gt}}}{\bigcup_{B}^{B^{gt}}} + \frac{\left|C - \bigcup_{B}^{B^{gt}}\right|}{|C|} \tag{6}$$

where $B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ is the ground-truth, and $B = (x, y, w, h)$ is the predicted box. $\frac{\bigcap_{B}^{B^{gt}}}{\bigcup_{B}^{B^{gt}}}$ is the Intersection over Union (IoU). C is the smallest box covering B and B^{gt} and $\frac{\left|C - \bigcup_{B}^{B^{gt}}\right|}{|C|}$ can be used to minimizes the distance between two central points. Focal loss function was used to resolve the class imbalance effect by reducing the loss for well-trained classes [34]. The loss function of the probability of hyperbola in this work is defined as:

$$Focal\ Loss\,(p_t) = -(1 - p_t)^{\gamma}\,\log(p_t) \tag{7}$$

The $-\log(p_t)$ is the cross entropy (CE) loss commonly used for classification tasks where the p is the probability for each category. The $(1 - p_t)^{\gamma}$ is the modulating factor added to address the data imbalance problem. The γ can be set in a range 0-5. In this work the γ was set to 2 to put more focus on an identified target.

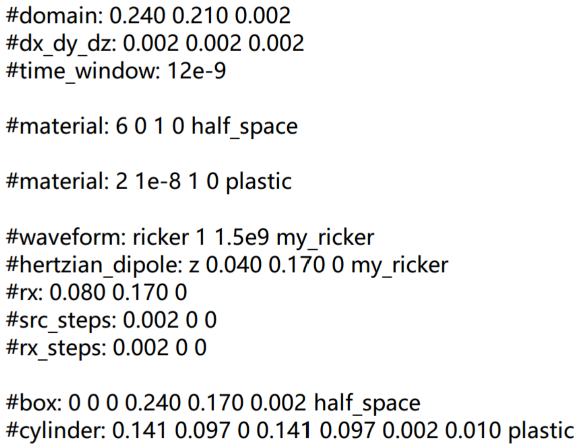
IV. EXPERIMENTAL EVALUATION

A. SIMULATED DATA

In this work, firstly to acquire some "real" data for the GANs model, GprMax, an open-source software to produce simulated electromagnetic wave propagation, was selected to generate simulated data. To simulate a buried PVE pipeline, we generate plastic cylinders with different diameters. The time window was set as 12×10^{-9}, and the ricker waveform as 1.5 GHz. To offer extra randomness, we applied randomly selected seed value for every iteration of generating training data. Thus, each image produced by GprMax was unique. A total of 80 A-scan traces comprised a single B-scan simulation. The diameter of a cylinder was used as classification label, which allowed us to condition the generator and identify the image during the GANs model training. Figure 6 shows a typical configuration file for GprMax.

B. REAL DATA

The trained hyperbola recognition model was evaluated on a group of real GPR B-scans images from rail infrastructure monitoring application provided by Railview [39]. The data was collected by a modular, off the shelf terrestrial based system named ZOND 12 with a 500MHz antenna. We conducted



```
#domain: 0.240 0.210 0.002
#dx_dy_dz: 0.002 0.002 0.002
#time_window: 12e-9

#material: 6 0 1 0 half_space

#material: 2 1e-8 1 0 plastic

#waveform: ricker 1 1.5e9 my_ricker
#hertzian_dipole: z 0.040 0.170 0 my_ricker
#rx: 0.080 0.170 0
#src_steps: 0.002 0 0
#rx_steps: 0.002 0 0

#box: 0 0 0 0.240 0.170 0.002 half_space
#cylinder: 0.141 0.097 0 0.141 0.097 0.002 0.010 plastic
```

FIGURE 6. GprMax configuration case.

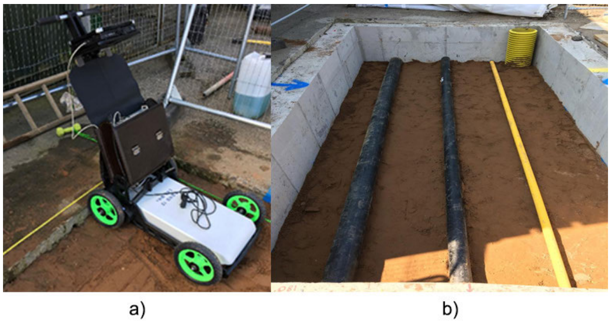


FIGURE 7. a) is the modular, off the shelf terrestrial based system (ZOND 12), b) is the on-site experimental environment.

the on-site experiment on a completion of the concrete surround measuring 4.8 metres in width × 2.4 metres length × 1.2 metres depth. Three plastic/rubber pipes with different diameter (180mm, 125mm, 63mm) were buried in the pit at a uniform depth of 800mm. Data was collected at 100mm intervals over a distance of 4.80 metres in width and the data collection time for this 11.52 Square meter test amounted to 53 minutes. The device and experimental environment were shown in Figure 7. Meanwhile, we have also used an additional real publicly available GPR dataset of 171 GPR B-scans images [40] to validate our model.

C. IMAGE ANNOTATION

As mentioned in the previous section, the bounding box of hyperbola in each image was annotated for object detection model training. In this work, 170 B-scan images were generated by GANs. Then the hyperbolas in each image were annotated manually by "LabelImg" [41], as shown in Figure 8.

D. THE PROPOSED GANs-BASED MODEL TRAINING

Figure 9 shows the process of training the GANs model. Adaptive Moment Estimation (Adam) [42] was used to replace RMSprop in the original method to speed up the convergence. In this work, the weighted sum of losses for the generator and the critic individually needs to be optimized. The algorithm starts with samples of real data and random noise vectors. For each iteration of the training loop and each generator update, the critic is updated a total of five times.

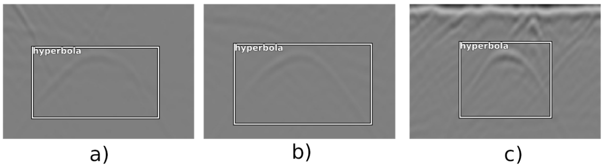
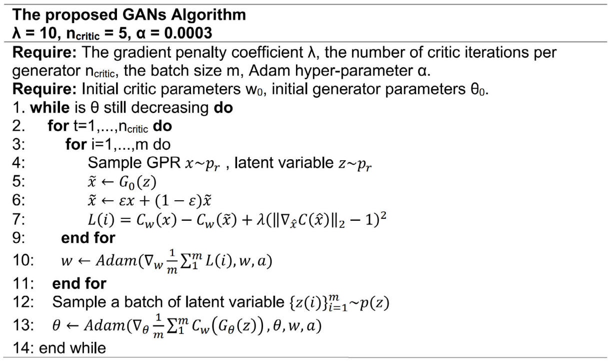


FIGURE 8. Annotated B-scan image.



```
The proposed GANs Algorithm
λ = 10, n_critic = 5, α = 0.0003
Require: The gradient penalty coefficient λ, the number of critic iterations per
generator n_critic, the batch size m, Adam hyper-parameter α.
Require: Initial critic parameters w_0, initial generator parameters θ_0.
while is θ still decreasing do
    for t=1,...,n_critic do
        for i=1,...,m do
            Sample GPR $x \sim p_r$ , latent variable $z \sim p_r$
            $\tilde{x} \leftarrow G_0(z)$
            $\tilde{x} \leftarrow \varepsilon x + (1-\varepsilon)\tilde{x}$
            $L(i) = C_w(x) - C_w(\tilde{x}) + \lambda(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2$
        end for
        $w \leftarrow Adam(\nabla_w \frac{1}{m}\sum_1^m L(i), w, a)$
    end for
    Sample a batch of latent variable $\{z(i)\}_{i=1}^m \sim p(z)$
    $\theta \leftarrow Adam(\nabla_\theta \frac{1}{m}\sum_1^m C_w(G_\theta(z)), \theta, w, a)$
end while
```

FIGURE 9. The algorithm of the proposed GANs.

The output and training parameters of each layer in the generator and the critic model are shown in Table 1 and Table 2 respectively.

E. OBJECT DETECTION MODEL TRAINING

The proposed object detection model was compiled and trained with the mmdetection (an open-source object detection toolbox based on PyTorch [43]). The trained Resnet50 by ImageNet was set as the backbone block in detection models. For the performance comparison, the most widely used multi-stage and single stage object detection models including Faster-RCNN [28], Cascade R-CNN [29], and SSD [33] were also implemented. The Stochastic Gradient Descent SGD) with a batch size of 8 was used for optimization. The learning rate (LR) was set as 0.001 in the first and it was finally decreased to 0.00001 following the training stage. In this work, each model was trained for 24 epochs. The object detection models configuration was shown in Table 3. As a multi-stage detection model, the Faster R-CNN and Cascade R-CNN have longer training time than the single stage detection models due to the Region Proposal Network (RPN).

TABLE 3. Object Detection Model Configuration.

Model Name	Backbone	Input Shape	Flops (GMac)	Training parameters (Million)	Train time (s/iter)
Faster R-CNN	Resnet50	512*512	63.66	41.53	0.353
Cascade R-CNN			91.29	69.17	0.523
SSD			58.22	36.04	0.265
YOLO V2			60.14	37.26	0.303
Our proposed model			61.22	37.74	0.285

As a single stage detection model, our proposed GPR detection model used the Feature Pyramid Networks to provide multi-scale features to improve the detection performance which slightly increased the model complexity. However, in the head part of our model, only three anchor boxes were used for each grid cell on each layer and processed in parallel for multi-scale features. Therefore, the training time is less than the traditional single stage detection models such as SSD and YOLO V2 in which normally five anchor boxes in their head parts were used. Here, anchor boxes represent a set of predefined bounding boxes of a certain height and width used to capture the scale and aspect ratio of specific object classes to be detected.

The mean Average Precision (**mAP**) was selected as the metric to evaluate the performance of the GPR detector. It is the mean of the Average Precisions (AP) for all classes. The AP is calculated as the average of maximum precision at 11 recall levels as follows, the AP_r (0) means the maximum precision when recall at value of 0.

$$AP = \frac{1}{11} \times \ (AP_r\ (0) + AP_r\ (0.1) + AP_r\ (0.2) + \ldots + AP_r\ (1)) \qquad (8)$$

If the AP matches the ground truth and Intersection over union (IOU) was above 0.5, then it was set as a correct value. We also use a metric called the inference time (fps), which indicates how many images can be inference per second.

F. EXPERIMENT EVALUATION

We have evaluated the proposed model in terms of performance and generalizability.

1) PERFORMANCE EVALUATION

Firstly, we have generated 170 GPR images based on our proposed GANS-based model.Figure 10 shows a set of samples of generated GPR B-scan images. The hyperbolas in these images show the desired behavior without showing much variation. There is no single noise distribution among hyperbolas and they are identical visually and the images are visually realistic.

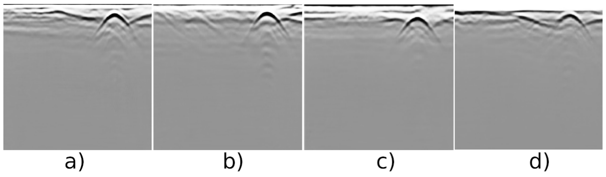


FIGURE 10. Realistic generated images.

Then, the object detection models were trained based on annotated images. Figure 11 shows the process of proposed model fitting.

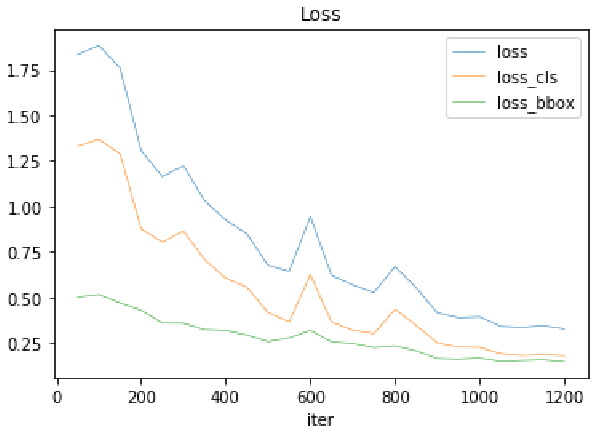


FIGURE 11. The loss plot for 1200 iteration during training the GRP Detection.

TABLE 4. Performance Comparison of Different Methods.

Model Name	mAP	Inf Speed (fps)
Faster R-CNN	0.95	22.6
Cascade R-CNN	0.96	20.5
SSD	0.90	24.4
YOLO V2	0.92	26.2
Our proposed model with MSE	0.95	27.2
Our proposed model with DIoU	0.97	27.2

Table 4 shows the model accuracy comparison. Faster R-CNN and Cascade R-CNN, as the most commonly used multi-stages detection models, their mAPs are 0.95 and 0.96 respectively, which show that these models work well for most hyperbolic signatures. Meanwhile, as a single-stage detection model, the mAP of our proposed GPR detection model achieves 0.95, which is close to the two state-of-the-art multi-stage detection models. By using the FPN as the neck part in the model architecture, our proposed model can detect the object on multi-scale of the deep features which can significantly improve its performance on small targets [44]. In this work, we have tried to replace the loss function of bounding box regression with DIoU. By directly minimizing the normalized distance of two central points, DIoU loss can achieve faster convergence speed and better performance. The mAP has significantly increased and achieved 0.97 by replacing the loss function with DIoU. The speed of inference is 29.5 image per second which is better than multi-stage detection models.

2) GENERALIZABILITY EVALUATION

To further evaluate the generalizability of the proposed detection model, we have conducted evaluation on the simulated data by GprMax and the GANs-based data by our model under different cross-validation strategies:

1) We have trained the model on GANs-based data and tested on GANs-based and simulated data, respectively.
2) Then we reverse the training and testing datasets by training the model on simulated data and testing on simulated and GANs-based data, respectively.

TABLE 5. The Model Performance on Independent Datasets.

Method	Training data	Evaluation data	mAP
Our Proposed model	GANs-based data	GANs-based data	0.97
	GANs-based data	Simulated data	0.95
	Simulated data	Simulated data	0.96
	Simulated data	GANs-based data	0.91

The simulated data are obtained from GprMax simulator. GANS-based data are obtained from our proposed GANs model. We have obtained the same number of images from GprMax simulator and from our GANS model respectively, 340 images in total. The results are show in Table 1 below. In the first group of the experiment, when training our proposed model with GANS data and testing it on both GANs-based and simulated data, the accuracies (mAP) are 0.97 and 0.95 respectively. In the second group of experiment, when training our proposed model with simulated data and testing it on both simulated data and GANs-based data, the accuracies (mAP) are 0.96 and 0.91 respectively. These results demonstrate the proposed model has good generalizability.

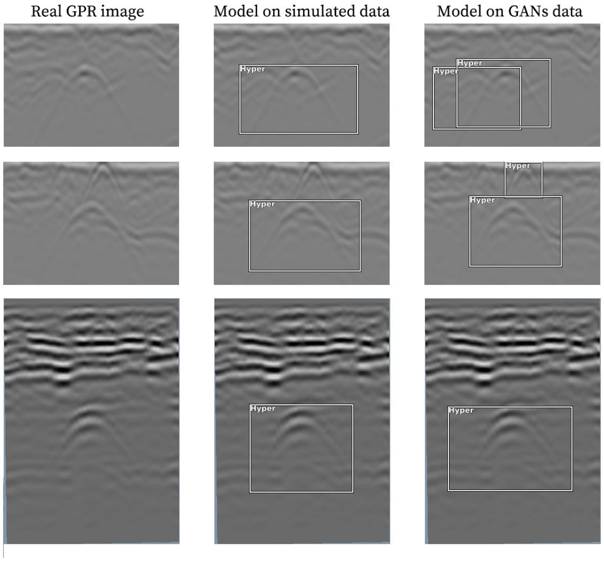


FIGURE 12. The examples of Hyperbola detection result. The first column shows three real raw GPR B-scans images. The second column shows the detection results based on the model trained on GprMax-based simulated data and the third column shows the detection results based on the model trained with GANs-based data.

Meanwhile, we have also tested the proposed model on collected real GPR data. Figure 12 shows the detection result. The first column shows three real raw GPR-sans sample images. The second column shows the detection results based on the model trained on GprMax-based simulated data and the third column shows the detection results based on the model trained with GANs data. It shows that the model trained on simulated data only recognizes one hyperbola in the first and second sample images, while the model trained on the GANs-based data recognizes all hyperbolas in the images. Both models have correctly identified hyperbolas in the third sample image.

V. CONCLUSION

In this work, we have proposed a GAN-based object detection tool consisting of two main components: a GANs-based model to enable generation of realistic GPR B-scan data and a single-stage object detection model with deep learning for automatically detect hyperbola in GPR B-scan image. The proposed approach has been evaluated with real data and has been compared with the state-of-the-art deep learning methods for object detection (i.e. Faster-RCNN, Cascade R-CNN, SSD and YOLO V2). The experimental results show that the proposed method outperforms the existing methods and achieved high accuracy of 97% for the mAP. Meanwhile, our proposed model shows good generalizability by a cross-validation on independent datasets. It has demonstrated the effectiveness of the proposed method in terms of addressing data scarcity and automatic detection of subsurface objects from GPR.

REFERENCES

[1] A. Giannopoulos, "Modelling ground penetrating radar by GprMax," *Construct. Building Mater.*, vol. 19, no. 10, pp. 755–762, Dec. 2005.

[2] C. Özdemir, Ş. Demirci, E. Yiğit, and B. Yilmaz, "A review on migration methods in B-scan ground penetrating radar imaging," *Math. Problems Eng.*, vol. 2014, Jun. 2014, Art. no. 280738.

[3] 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, Jan. 2014.

[4] 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 Sens.*, vol. 8, no. 5, p. 430, May 2016.

[5] Y. Djenouri, G. Srivastava, and J. C.-W. Lin, "Fast and accurate convolution neural network for detecting manufacturing data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2947–2955, Apr. 2021.

[6] L. E. Besaw and P. J. Stimac, "Deep convolutional neural networks for classifying GPR B-scans," *Proc. SPIE*, vol. 9454, May 2015, Art. no. 945413. [Online]. Available: http://adsabs.harvard.edu/abs/2015SPIE.9454E.13B

[7] S. Lameri, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro, "Landmine detection from GPR data using convolutional neural networks," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 508–512.

[8] 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 *Proc. 9th Int. Workshop Adv. Ground Penetrating Radar (IWAGPR)*, Jun. 2017, pp. 1–5.

[9] C. Warren, A. Giannopoulos, and I. Giannakis, "gprMax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar," *Comput. Phys. Commun.*, vol. 209, pp. 163–170, Dec. 2016.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[11] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*. [Online]. Available: http://arxiv.org/abs/1511.05440

[12] J. Liu, F. Qu, X. Hong, and H. Zhang, "A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 3877–3888, Jul. 2019.

[13] H. Harkat, Y. Elfakir, S. D. Bennani, G. Khaissidi, and M. Mrabti, "Ground penetrating radar hyperbola detection using scale-invariant feature transform," in *Proc. Int. Conf. Electr. Inf. Technol. (ICEIT)*, May 2016, pp. 392–397.

[14] 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, pp. 491–504, Aug. 2016.

[15] G. Terrasse, J.-M. Nicolas, E. Trouve, and E. Drouet, "Automatic localization of gas pipes from GPR imagery," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 2395–2399.

[16] C. Maas and J. Schmalzl, "Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar," *Comput. Geosci.*, vol. 58, pp. 116–125, Aug. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S009830041300112X

[17] 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, Mar. 2014.

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: http://arxiv.org/abs/1710.10196

[19] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," 2016, *arXiv:1610.07584*. [Online]. Available: http://arxiv.org/abs/1610.07584

[20] C. Veal, J. Dowdy, B. Brockner, D. T. Anderson, J. E. Ball, and G. Scott, "Generative adversarial networks for ground penetrating radar in hand held explosive hazard detection," *Proc. SPIE*, vol. 10628, Apr. 2018, Art. no. 106280T. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10628/106280T/Generative-adversarial-networks-for-ground-penetrating-radar-in-hand-held/10.1117/12.2307261.short

[21] T. Truong and S. Yanushkevich, "Generative adversarial network for radar signal synthesis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7.

[22] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016, *arXiv:1611.02163*. [Online]. Available: http://arxiv.org/abs/1611.02163

[23] K. Aderghal, M. Boissenin, J. Benois-Pineau, G. Catheline, and K. Afdel, "Classification of sMRI for AD diagnosis with convolutional neuronal networks: A pilot 2-D+ϵ study on ADNI," in *MultiMedia Modeling* (Lecture Notes in Computer Science), L. Amsaleg, G. Guðmundsson, C. Gurrin, B. Jónsson, and S. Satoh, Eds. Cham, Switzerland: Springer, 2017, pp. 690–701.

[24] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*. [Online]. Available: http://arxiv.org/abs/1704.00028

[25] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: http://arxiv.org/abs/1805.08318

[26] J. Feng, L. Yang, H. Wang, Y. Song, and J. Xiao, "GPR-based subsurface object detection and reconstruction using random motion and DepthNet," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7035–7041.

[27] X. Xu, Y. Lei, and F. Yang, "Railway subgrade defect automatic recognition method based on improved faster R-CNN," *Sci. Program.*, vol. 2018, Jun. 2018, Art. no. 4832972.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[29] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[32] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 21–37.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[35] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[36] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853

[37] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," 2019, *arXiv:1902.09630*. [Online]. Available: http://arxiv.org/abs/1902.09630

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," 2019, *arXiv:1911.08287*. [Online]. Available: http://arxiv.org/abs/1911.08287

[39] *R. Ltd. Railview*. Accessed: Jan. 5, 2021. [Online]. Available: https://www.railview.co.uk/

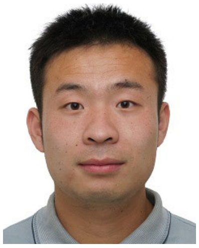
[40] I. X. Chen. (Jul. 2020). *Irenexychen/Gpr-Data-Classifier*. Accessed: Jun. 2018. [Online]. Available: https://github.com/irenexychen/gpr-data-classifier

[41] Tzutalin. (Mar. 2020). *LabelImg*. Accessed: Sep. 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[43] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*. [Online]. Available: http://arxiv.org/abs/1906.07155

[44] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, *arXiv:1612.03144*. [Online]. Available: http://arxiv.org/abs/1612.03144



XIN ZHANG received the B.S. degree from the PLA Academy of Communication and Commanding, China, in 2009, and the Ph.D. degree in cartography and geographic information system from Beijing Normal University(BNU), China, in 2014. He is currently an Associate Researcher with Manchester Metropolitan University (MMU). His current research interests include remote sensing image processing and deep learning.

LIANGXIU HAN received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2002. She is currently a Professor of computer science with the Department of Computing and Mathematics, Manchester Metropolitan University. Her research interests include the development of novel big data analytics and development of novel intelligent architectures that facilitates big data analytics (e.g., parallel and distributed computing, and cloud/service-oriented computing/data intensive computing) and applications in different domains using various large datasets (biomedical images, environmental sensor, network traffic data, and web documents). She is currently a principal investigator or Co-PI on a number of research projects in the research areas mentioned above.

MARK ROBINSON is currently a Professor of rail systems engineering and also the Group Lead Future Mobility and the Director of NewRail the Centre for Rail Research, School of Engineering, Newcastle University. He is also a specialist in infrastructure and rail vehicle inspection and monitoring. His current research interests include asset management, infrastructure and rail vehicle maintenance, and railway technology and inspection and monitoring techniques.

He has been actively involved in UIC International Rail Research Advisory Board IRRB, since 2011, and has been the Vice Chairman since 2017. In January 2015, he was elected as the President of the European Conference of Transport Research Institutes (ECTRI), from January 2015 to January 2017. In December 2016, he was re-elected President for his final term to January 2019. From 2014 to 2017, he was honored by the China High End Foreign Experts Programme. In 2016, he was awarded 100 Talents from Fujian Province, both for high rail research activities. He is currently a Fujian Province High End Expert Professional.

ANTHONY GALLAGHER is currently pursuing the M.Sc.Tech. degree in photonics and optoelectronics from the School of Physics, University of New South Wales, Sydney, Australia. He is currently the Chief Scientist of Railview Ltd. He has international experience in Singapore, Australia, and Canada when establishing an advanced radar research laboratory in Edmonton, Canada, equipped with a full array of radio frequency test equipment. He was also involved in the development of synthetic aperture radar signal processing algorithms to implement volumetric (3D) imaging of objects and designed transceiver hardware for the implementation of volumetric imaging radar. His experience in research extends to conducting research on radio frequency and visible spectrum active and passive remote sensing systems for unmanned aerial vehicles, which also involved business plan creation, production of technical reports, and grant applications. He has also conducted research on synthetic aperture radar systems for small satellites; researched computational techniques such as the time-domain discrete dipole approximation to model radar backscatter from complex terrain and investigated the effects of curvilinear vehicle motion on synthetic aperture image formation.