

A CNN model for predicting size of buried objects from GPR B-Scans

Nairit Barkataki^{a,*}, Banty Tiru^b and Utpal Sarma^a

^aDepartment of Instrumentation & USIC, Gauhati University, Guwahati, India

^bDepartment of Physics, Gauhati University, Guwahati, India

ARTICLE INFO

Keywords:

deep learning

CNN

object size prediction

ground penetrating radar

ABSTRACT

A convolutional neural networks (CNN) model for predicting size of buried objects from ground penetrating radar (GPR) B-Scans is proposed. As a pre-processing step, Sobel, Laplacian, Scharr, and Canny operators are used for edge detection of the hyperbolic features. The proposed CNN architecture extracts high level signatures in the initial stages of the model and learns additional low-level features when the input data passes through the neural network to finally make an estimation of the required parameter. Artificially generated GPR B-Scans are used to train the model. The proposed method demonstrates good performance in predicting buried object size. Upon comparison, Scharr operator followed by a deep CNN model showed the best performance, having the minimum mean absolute percentage error of 6.74 when tested on new, unseen data.

1. Introduction

Ground penetrating radar (GPR) is a nondestructive technique (NDT) that can be used for various applications like archaeology, geological surveys, landmine detection, detection and mapping of subsurface utilities, detection of subsurface structures such as tunnels and boreholes, etc.

Ever since the GPR technology came to force, the basic principles used in creating GPR images have remained more or less the same. A GPR transmits pulses of electromagnetic waves at various frequencies depending on the choice of the user and the type of application. Due to different dielectric properties, the boundaries of buried objects and soil layers reflect radar waves back to a receiving antenna. The resulting radargram (GPR image) is an image of the subsurface that shows the various regions of heterogeneity. The data collected during the survey needs to be processed afterwards to improve and interpret the assessment. The hyperbolic feature is the most common shape found in B-Scan images of GPR. It is a challenging task and requires highly skilled people to properly identify and classify the various hyperbolic features in a GPR image. Additionally, corruption of GPR data due to noise creates added hurdles in the interpretation of GPR data. Therefore, GPR data processing continues to be a challenge to the scientific community.

1.1. Literature Review

In recent years, a variety of methods have been proposed by researchers for the interpretation of GPR data. Torrione et al. (2013) successfully applied the Histograms of Oriented Gradients (HOG) feature extraction technique to landmine detection in GPR data to detect landmines and proposed that other image processing techniques may also be applied for detection of targets in GPR data. A hyperbola fitting technique was used by Wahab et al. (2013) to estimate the radius of buried pipes and cables. Ko et al. (2012) used Fourier Transform and Principal Component Analysis to detect landmines in various burial conditions.

Considerable research has been done in the detection of hyperbolic features and determination of regions of interest in GPR images for further interpretation of various parameters. Different strategies have been used on the topic of hyperbola detection. Fitting algorithms like Radon (Dell'Acqua et al. (2004)), wavelet (Windsor et al. (2005)) and Hough (Chen and Cohn (2010); Liu et al. (2019)) have been used for feature extraction in GPR images. Other feature extractors like Sobel operators (Li et al. (2016)) and Canny operators (Mertens et al. (2015); Bugarinović et al. (2020)) are also used as a pre-processing step. Statistical approaches like least square (Chen and Cohn (2010)), fuzzy shell clustering (Delbo et al. (2000)), moment of inertia optimization (Chaudhuri and Samanta (1991)) etc., have also been

*Corresponding author

✉ nairitb@gauhati.ac.in (N. Barkataki)

ORCID(s): 0000-0002-7988-072X (N. Barkataki); 0000-0003-1761-0983 (B. Tiru); 0000-0001-7888-1088 (U. Sarma)

used to extract hyperbolic features. These methods usually involve multi-stage techniques. Initially, a region of interest is determined and then a hyperbola fitting algorithm is applied to it. All these methods are computationally slow but perform well from the point of view of accuracy and sensitivity.

Maas and Schmalzl (2013) proposed a multi-stage technique which could recognise and fit hyperbolae in real time. They adapted OpenCV library to localise reflection hyperbolas in GPR B-Scans. The developed algorithm showed promising results when applied on using realistic data. They used Canny operator for edge detection and Hough transform for hyperbola fitting. Later, Dou et al. (2016) working on a similar problem, proposed a novel technique for real-time hyperbola recognition by using a neural network classification algorithm. They used synthetic input images of size 100×100 pixels. Due to the fast computation speed, both methods proposed by Maas and Schmalzl (2013) and Dou et al. (2016) are suitable for real-time applications.

However, studies by Jacob and Urban (2016) have shown that under certain conditions, Common Midpoint (CMP) method appeared to be a more accurate way to determine GPR subsurface velocity than hyperbola fitting. Zhao and Al-Qadi (2016) applied CMP method when estimating the thickness of asphalt.

Machine learning algorithms can be used for meaningful interpretation of GPR data. Machine learning in GPR data interpretation has many applications, some of which include detection of landmines (Smitha and Singh (2020); Frigui and Gader (2008)), predicting geometry and material classification of buried objects (Syambas (2012); Lu et al. (2014)), automatic detection and interpretation of GPR data (Shihab et al. (2002)), etc. However, with the increase in amount of data, detection of target features takes a long time. Other approaches use machine learning methods to concentrate on hyperbola detection as a first step, followed by a fitting algorithm. Ristić et al. (2017) used an artificial neural network in extraction of hyperbolic features and segregation of segments from a radargram.

Neural networks are gradually becoming popular since they were first introduced in the 1980s. Machine learning algorithms look at the data as a whole and draw decision boundaries around the samples. Since they treat all given features in a similar way, any change in even an insignificant feature may cause degradation in the model's performance. A deep learning model, on the other hand, contains multiple layers of neural networks which work on different aspects of the input data. It even checks the significance of different features and tries to give lower weightage to insignificant ones.

For automatic detection of characteristic signatures of buried objects, neural network (NN)-based methodologies were developed by Al-Nuaimy et al. (2000) and Gamba and Lossani (2000). The application of multi-layer perceptron (MLP) neural network was effective in automatic identification and location of embedded steel reinforcing bars in concrete (Shaw et al. (2005)).

In recent years, extensive studies have been carried out on interpretation techniques based on deep-learning-based models. This has brought to light the many advantages of deep learning techniques. With increase in GPR data, the performance of traditional machine learning techniques suffer. This is where deep learning techniques have an edge over the traditional machine learning techniques (Najafabadi et al. (2015)). Moreover, since GPR data contains considerable environmental and experimental noise, deep learning techniques are generally better in data interpretation. Over time, researchers have started using CNN-based hyperbolic pattern recognition methods (Krizhevsky et al. (2012); Besaw and Stimac (2015); Lameri et al. (2017)) for classification of objects.

With the advent of simulation tools like gprMAX (Warren et al. (2016)), researchers have started using artificially generated GPR data for machine learning based classification of underground objects (Lu et al. (2014); Zhang et al. (2016); Kafedziski et al. (2018); Giannakis et al. (2021)).

Recently, models based on deep CNN were used in pattern recognition (Ding et al. (2016)) and classification (Chen et al. (2016)) of radar targets. Compared to a conventional neural network, the architecture of the deep CNN consists of 2 or more pairs of convolution and pooling layers. Deep CNN improves the accuracy of the target detection in GPR data. Tong et al. (2018) employed multi-stage CNNs to automatically classifying subgrade defects from X-ray computed tomography images, while Jiang et al. (2018) used a multi-stage CNN to identify asphalt mixtures from GPR data. Although, various techniques have reported improved performance of CNNs, they have not been extensively used in interpretation of GPR data.

Wiwatrojanagul et al. (2017) proposed a new method to automatically locate rebars in reinforced concrete structures. They used a processing technique to locate the rebar and to estimate the wave propagation velocity using hyperbolic signature from GPR data. The proposed method could estimate both the cover thickness and locate the buried objects with a high degree of accuracy. Later, Liu et al. (2020) worked on similar lines and proposed a method which used deep learning to automatically detect and localise buried targets. Single Shot Multibox Detector (SSD) was used to identify the regions of interest in GPR images which consisted hyperbolic signatures. This study concluded that the

proposed approach could detect buried rebars in real time with a high degree of accuracy.

Most of the above methods concentrate on the detection and the localisation of buried objects, and the classification of such objects based on their shapes and compositions. However, very few researchers have worked on prediction of size of underground objects. Those that have dealt with prediction of size have used extrapolation (Syambas (2012)) and gaussian regression techniques (Pasolli et al. (2009)).

1.2. Aim of Present Work

In this paper, a feature-based algorithm for object size prediction in GPR data is proposed. It uses deep Convolutional Neural Network (CNN) estimator to decrease error and increase speed of prediction of size of targets from GPR images. Feature extraction is performed on the original pattern of GPR images prior to training the dataset in the proposed CNN model. Many training images are required for successful training of the model and to improve accuracy. To the best knowledge of the authors, no prior work has been done using hyperbolic feature extraction as a pre-processing step followed by a CNN-based estimator to predict size of buried objects.

2. GPR Data

The electromagnetic simulator gprMax is used to generate simulated data for training the proposed model. Further, acceleration of the simulations is achieved by using NVIDIA's CUDA framework (Warren et al. (2019)).

To create a model for simulation, several parameters need to be set. Figure 1 shows one of the models created using gprMax.

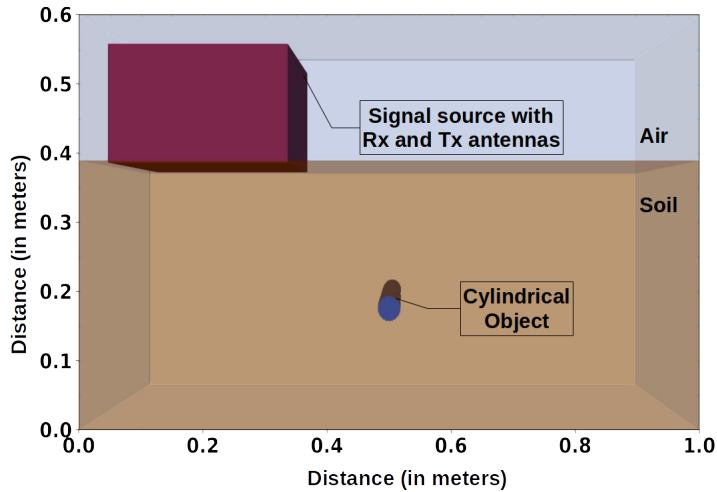


Figure 1: Simulated model

The dimensions of the model are $1000 \times 400 \times 600$ ($X \times Y \times Z$) mm^3 . There is a 210 mm layer of air at the top and a soil layer of 390 mm below it. In Figure 1, a cylindrical object, made of aluminium ($\sigma = 3.5 \times 10^7$ Siemens/metre, $\epsilon_r = 10.8$), is located 213 mm below the ground. The signal source along with the transmitter (Tx) and receiver (Rx) antennas are placed above the soil layer.

1000 GPR B-Scans are created for 7 soil types with random values each for object depths and radii. The dielectric parameters for the soil types are given in Table 1 (Von Hippel (1954)).

Depth values are taken randomly between 10 mm and 215 mm below the soil surface. As for radius of the target, random integer values between 5 mm and 20 mm are selected. Care is taken to automatically adjust the depth of the target for larger values of radius, so that the target is always below the soil surface. The parameters used in the simulation are given in Table 2.

The value of time window depends mainly on soil properties (σ & ϵ_r) and the depth of the buried object. Its value should be large enough to allow receipt of the echo signal by the receiver. The maximum value of time window is found to be 14 ns and this value is taken for all simulations.

The simulations are accelerated using NVIDIA RTX3090 GPU. For each model, 108 A-Scans are taken, sufficient to capture the hyperbola created by the EM wave reflection from the target. Figure 2 shows a sample B-Scan image

Table 1
Properties of Soil Types

Sl. No.	Soil Type	Conductivity σ (S/m)	Relative permittivity ϵ_r
1	Dry and sandy	0.002	10
2	Marshy land, densely wooded	0.0075	12
3	Agricultural land	0.01	15
4	Rocky land, steep hills	0.002	12.5
5	Hilly terrain (to about 1000 m)	0.001	5
6	Pastoral Hills, rich soil	0.007	17
7	Pastoral medium hills and forestation	0.005	13

Table 2
Simulation parameters

Parameters	Parameter values
Source Frequency	400 MHz
Source Waveform	Gaussian
Resolution (Spatial)	2 mm
A-Scan intervals	6 mm
No of A-Scans	108
Time Window	14 ns

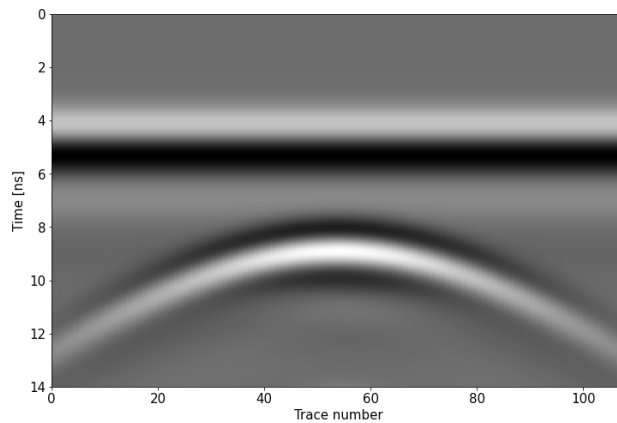


Figure 2: A simulation result

which is composed of 108 A-Scans. An object of radius 18 mm (36 mm diameter) is buried 213 mm below the soil surface. Each B-Scan file has 3636×108 data points. All 1000 B-Scan files are stored in a three dimensional (3D) numpy array along with their corresponding attributes (object radius). A representation image of the array of stacked B-Scan files is shown in Figure 3.

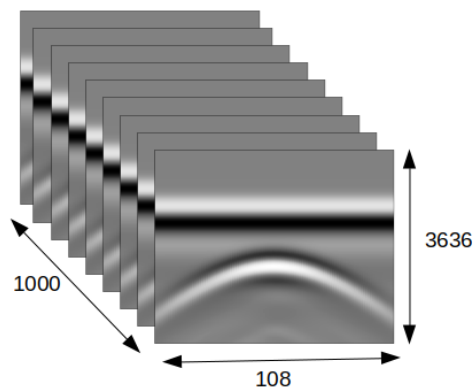


Figure 3: Image data is stacked in a 3D array

3. GPR Principle

As can be seen in Figure 4, the GPR receives multiple reflections from a buried target as it moves across the scanning direction.

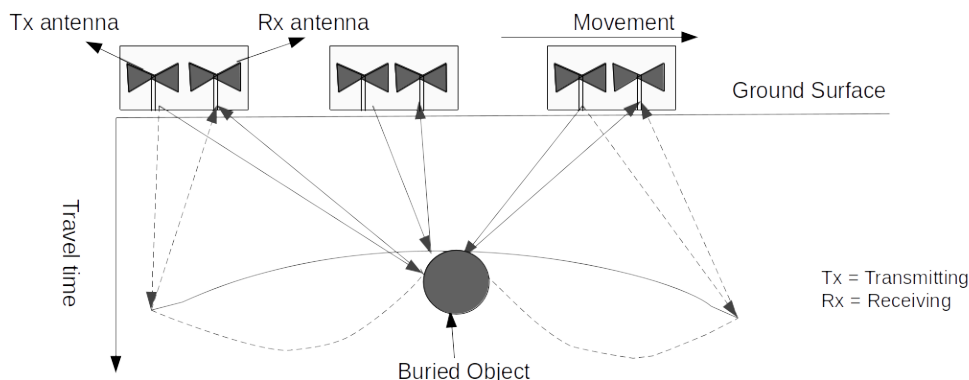


Figure 4: Hyperbola formation due to reflections from buried target

The larger the size of the buried object, the greater will be the area from which reflections occur. When presenting the effect of the diameter of buried object on hyperbolic reflections, *Wiwatrojanagul et al. (2017)* observed that the amplitude of the peak hyperbolic reflection has a good correlation with the diameter of the buried objects. Later, in the findings reported by *Liu et al. (2020)*, it was seen that the shape of the hyperbolic curve changed slightly with different rebar diameters, even though it is more sensitive to the rebar depth than the rebar diameter. The input image of the proposed model contains 108 A-Scans taken at 6 mm intervals. This indirectly contains size related information regarding the hyperbola. This should help in extracting different characteristic features from the hyperbolic image.

4. Convolution Neural Networks

Conventional computer vision methods have been applied for structural analysis (*Torrione et al. (2013)*; *Wahab et al. (2013)*; *Koch et al. (2015)*; *Zakeri et al. (2017)*), detection of concrete cracks (*Liu et al. (2013)*), recognition and visualization of cracks (*Adhikari et al. (2014)*), measurement of road surface distresses (*Zhang and Elaksher (2012)*), geological parameterization (*Liu and Durlofsky (2020)*), etc. The techniques applied in the studies mentioned above require large amount of image pre-processing when the training dataset is being prepared. The training process is tedious as well. CNN models typically perform feature extraction through a series of convolutional layers, pooling layers, and fully connected hidden layers. CNN has shown exceptional performance in various domains. Its use in computer vision is well known with much work being done on face recognition with varying illuminations and facial expressions among others. CNN architectures have also been implemented on portable devices like FPGAs for various applications (*Guo et al. (2017)*; *Nguyen et al. (2019)*). The pre-processing required in a CNN model is lower than that of algorithms (*Cheng and Wang (2018)*; *Krizhevsky et al. (2012)*). The architecture of a typical CNN binary classifier with an input image size of 128×128 is shown in Figure 5.

CNNs used in practical purposes have multiple pairs of convolutional and pooling layers. The output of the final pooling layer is connected to a hidden layer, the output of which is passed to a final classification/estimation layer.

5. Proposed CNN Architecture

A CNN extracts high-level features such as edges from the input image and reduces the dataset to a form which is easier to process. The spatial size of the convolved feature is reduced by the pooling layer. This decreases the computational requirements for processing the data. It is also useful for extracting dominant features.

Before feeding it to a neural network, the output from the final pooling layer is flattened into a column vector.

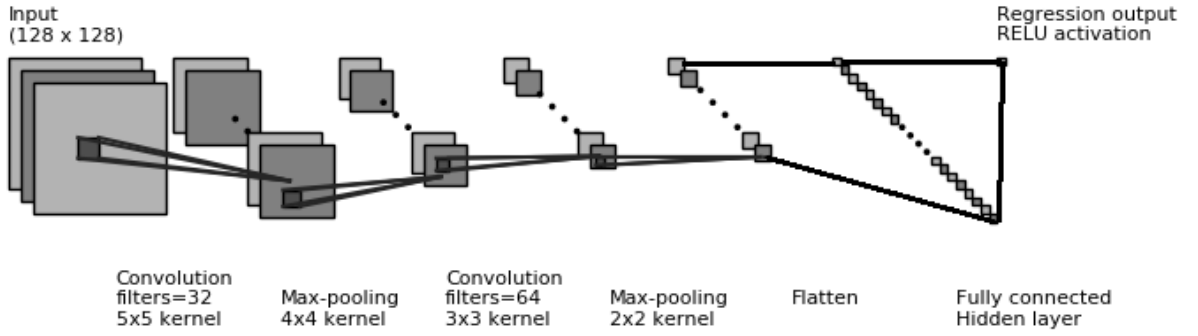


Figure 5: Structure of a generic CNN architecture

5.1. Feature Extraction

The overall GPR data processing pipeline in this work consists of feature extraction, splitting of dataset into train-test sets, normalising the data, training of CNN model, and validation.

As a means of feature extraction, edge detection is used. There are several operators for edge detection that are used in image processing. In this work, Sobel, Laplacian, Scharr, and Canny operators are used for comparing the performance of the proposed CNN architecture. Hyperbolic curves are one of the distinguishing features in GPR images. The parameters of a hyperbolic signature can be used to estimate the size and location of the related target object in addition to the characteristics of the medium (e.g.: soil). In edge detection we try to find the regions in an image where there is a sharp change in intensity or colour. The higher the value, steeper the change and vice versa.

5.1.1. Sobel Operator

A very common operator for edge detection is a Sobel operator. Sobel is a very popular operator and is widely used by researchers as an edge detection tool (Chen et al. (2020)). The Sobel operator is a gradient based method and it calculates the first derivatives of the image separately for the X and Y axes. The derivative in X direction is computed by convolving the image $f(x, y)$ with a kernel G_x . For a kernel size of 3, G_x is computed as shown in equation 1. The derivative in Y direction is computed as shown in equation 2.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * f(x, y) \quad (1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * f(x, y) \quad (2)$$

From equations 1 and 2, the gradient of an image $f(x, y)$ with respect to x and y directions can be shown as given in equation 1. The gradient will be in whichever direction the change will be more intense. This gradient direction is calculated as given by equation 2. Finally, at each pixel of the image, the approximation of the gradient is calculated by equation 3.

$$\nabla f = [G_x, G_y] = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \quad (3)$$

$$\theta = \tan^{-1} \left(\frac{G_x}{G_y} \right) = \tan^{-1} \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right) \quad (4)$$

$$G = \|\nabla f\| = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (5)$$

In this work, 7×7 Sobel kernels are implemented. By increasing the size of the convolution matrix, the edge detector becomes less sensitive to local noise. This is useful for low contrast images where edges might not be easy to detect. Sobel operator has the advantage of providing edge detection with simultaneous reduction in noise (gaussian smoothing).

5.1.2. Laplacian Operator

The second edge detection method used to compare the CNN performance is the Laplacian operator (Bergholm (1987)). However, unlike the Sobel operator, the Laplacian uses only one kernel. Laplacian operator is computationally faster than Sobel operator and it calculates second order derivatives in a single pass.

In this work, 5×5 Laplacian kernels are implemented due to the advantages mentioned previously.

5.1.3. Scharr Operator

The third method used for CNN performance comparison is the Scharr operator which calculates the second derivatives in both X and Y axes.

In this work, 5×5 Scharr kernels are implemented due to the advantages mentioned previously.

5.1.4. Canny Operator

The fourth method used for CNN performance comparison is the Canny operator. It has been used as a pre-processing step for automated detection of hyperbolic features (Bugarinović et al. (2020)). Di and Gao (2014) demonstrated the use of Canny edge detection for 3D seismic discontinuity enhancement. It is a multi-stage edge detector algorithm which goes through the following steps:

- i. Noise reduction using a Gaussian filter.
- ii. Calculate intensity gradient using Sobel filter, using equations 4 and 5.
- iii. Remove unwanted pixels which may not be part of the edges (non-maximum suppression).
- iv. Using high and low threshold intensity values, focus on the edges which are important for the present application and filter out the unwanted pixels which non-maximum suppression could not.

In this work, a Canny operator having 20 as low threshold and 120 as high threshold is used. The threshold values are determined using trial-and-error method for best validation loss during CNN training.

The fundamental concept in all the four feature extraction techniques is that the targets (buried object) produce hyperbolic signatures in GPR B-Scans which can be summarised in terms of size and orientation of the hyperbolic shapes.

5.2. Training of CNN model

The dataset containing 1000 B-Scans (108,000 A-Scans) is split so that 900 files are used for training and 100 for validation. Care is taken so that we get the same split for each edge detector used. The data is then normalised separately for train and test sets and reshaped to add a 4^{th} dimension for colour channel (1 in this case since it is greyscale).

5.2.1. Convolution Layer

A convolution layer applies a filter to an input image which results in an activation. The corresponding activation function decides which nodes get activated. Repeated application of a filter results in a feature map. Multiple filters are used in parallel for each convolutional layer. This helps it to extract multiple features in parallel. The Rectified

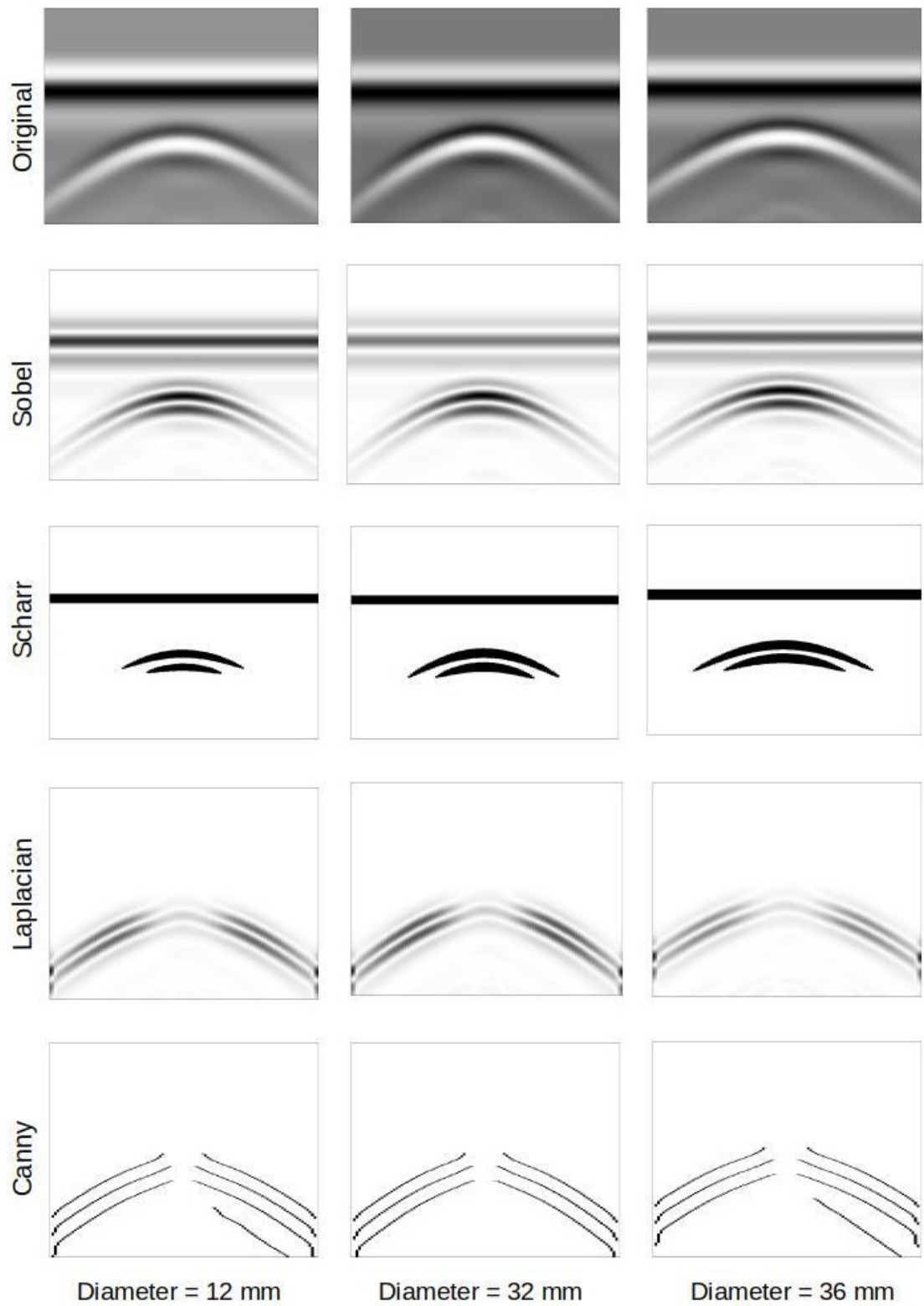


Figure 6: Output of filters used in object of different sizes

Linear Unit (ReLU) activation function is used in both the convolutional and hidden layers. ReLU is a non-linear activation function used in deep or multi-layer neural networks. It can be represented as:

$$f(x) = \max(0, x) \quad \text{where, } x \text{ is an input value.} \quad (6)$$

The output of ReLU is equal to 0 when the input value is less than 0 and equal to the input value when the input is positive. Thus, equation 6 can be re-written as:

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \quad (7)$$

5.2.2. Pooling Layer

Of the two types of pooling available, Max Pooling downsamples the input image by computing the maximum value from the portion covered by the kernel. On the other hand, Average Pooling computes the average from the portion covered by the kernel. In most cases, Max Pooling has a better performance than Average Pooling. The convolutional and pooling layers, together form an i-th layer of the CNN.

5.2.3. Fully Connected Layer

A fully connected layer is an important final stage of any convolution network. Here, the output of the network's last convolution layer are flattened to a single 1 dimensional vector. The input nodes of the fully connected layer have a complete connection to all the activations from the previous flattened layer.

5.2.4. Loss Function

The mean absolute percentage error (MAPE) is used as a loss function for training the CNN model. It is the mean of the absolute differences between our target and predicted values. It is defined as the mean of absolute relative errors:

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{P_t - A_t}{A_t} \right| \quad (8)$$

where N is the number of samples, (P_t) is the predicted data value and (A_t) is the actual data value.

5.2.5. Optimization

The use of an appropriate optimizer is essential in improving the model's performance. Optimizers minimise losses by adjusting the hyper-parameters of the network like weight and learning rate. Stochastic Gradient Descent (SGD) was perhaps the most popular optimizer during the initial days of deep learning. Over the years, many improvements on SGD like momentum and Nestorov (Botev et al. (2017)) have been reported. Adaptive optimizations like AdaDelta, RMSProp, and Adam (Adaptive Moment Estimation) have become popular recently as they adjust the learning rates of the parameters in different proportions, resulting in faster convergence and smoother training process. Adam has become quite popular because it converges faster than the other adaptive optimizers. Hence, it is used in optimizing the proposed CNN model's training process with an initial learning rate of 0.0001.

5.2.6. Evaluation Metrics

MAPE is used as a metric to measure the performance of the CNN training and validation. It is a measure of prediction accuracy in statistics. MAPE is commonly used for checking the performance of regression models. MAPE can be expressed as a percentage, making it understandable to all types of audiences. By contrast, other metrics that are not expressed in relative terms or as percentages usually require domain expertise and context to understand the significance of their numerical values.

5.3. Final Architecture

After multiple training sessions, it is found that 4 filters of 5x5 kernel size in the first Convolutional layer gives the best MAPE, irrespective of the number of hidden layers in the model. Hence, this is taken as the base for the model, after which several hidden layers are added. Performance is evaluated for each addition of hidden layer. Table 3 shows the validation losses corresponding to different numbers of hidden layers used, for various edge detectors.

Table 3
Validation Losses for different numbers of hidden layers

Sl. No.	Number of Hidden Layers	Validation Loss (MAPE)				
		No Edge Detector	Sobel	Laplacian	Scharr	Canny
1	1 Layer	28.99	33.1	30.41	8.70	18.51
2	2 Layers	29.59	30.40	31.91	9.55	19.96
3	3 Layers	30.84	34.46	34.27	7.72	20.16
4	4 Layers	30.32	35.68	32.99	7.65	21.31
5	5 Layers	16.39	31.23	29.29	6.94	23.97
6	6 Layers	16.34	30.56	29.25	8.71	25.58
7	7 Layers	16.14	30.24	-	8.35	-
8	8 Layers	16.55	-	-	8.36	-
9	9 Layers	16.11	-	-	8.50	-
10	10 Layers	20.49	-	-	9.87	-

Table 4

Estimated diameters (mm) of buried targets by the CNN model

Sl. No.	True value	Best Prediction	Worst Prediction	Median
1	10	9.58	7.27	9.51
2	12	13.16	17.48	14.59
3	14	14.18	17.18	14.18
4	16	15.33	17.12	15.33
5	18	19.06	21.39	20.11
6	20	19.53	23.45	20.95
7	22	22.30	24.57	21.80
8	23	24.51	24.85	24.68
9	24	23.93	30.25	24.51
10	26	26.05	23.70	25.73
11	27	32.06	32.06	32.06
12	28	30.59	23.46	27.32
13	30	29.47	23.56	30.81
14	31	32.68	35.58	34.13
15	32	33.33	24.89	33.73
16	34	35.09	29.33	35.09
17	36	36.30	39.74	36.30
18	38	39.70	27.72	34.86
19	39	36.28	36.28	36.28
20	40	40.40	37.80	39.24

The architecture of the best CNN model is shown in Figure 7. The input convolution layer is followed by a MaxPool layer of pool size 2×2 . 32 filters of kernel size 3×3 is used in the 1st hidden layer followed again by a MaxPool layer of pool size 2×2 . Another similar pair of convolution-pooling layer is placed after that, with the number of filters being 64 in this case. Three more convolution layers are used after that with filter numbers 64 (3×3), 64 (3×3), and 66 (2×2) respectively. At this point, the outputs are flattened to a 1D matrix and a fully connected hidden layer of 64 neurons is added. A 1-way ReLU regression layer is used after the hidden layer, to estimate the required parameter from each input image. Mini batch gradient descent method of batch size 32 is used for training the CNN model. The CNN model trains over a series of epochs and makes an estimation of the required parameter, i.e., diameter ($2 \times$ radius) of the buried object. MAPE (loss function) during validation is monitored and early stopping callback API of Keras is used to stop the training if the model performance does not improve after 100 successive epochs.

NVIDIA RTX3090 GPU was used in CNN training. Using a GPU, the CNN training time reduced from several hours on CPU to less than 1 hour. The whole GPR data processing pipeline, including the CNN and its optimisation algorithms, is written in Python, along with its image processing extension OpenCV and its deep learning extensions, Keras and TensorFlow.

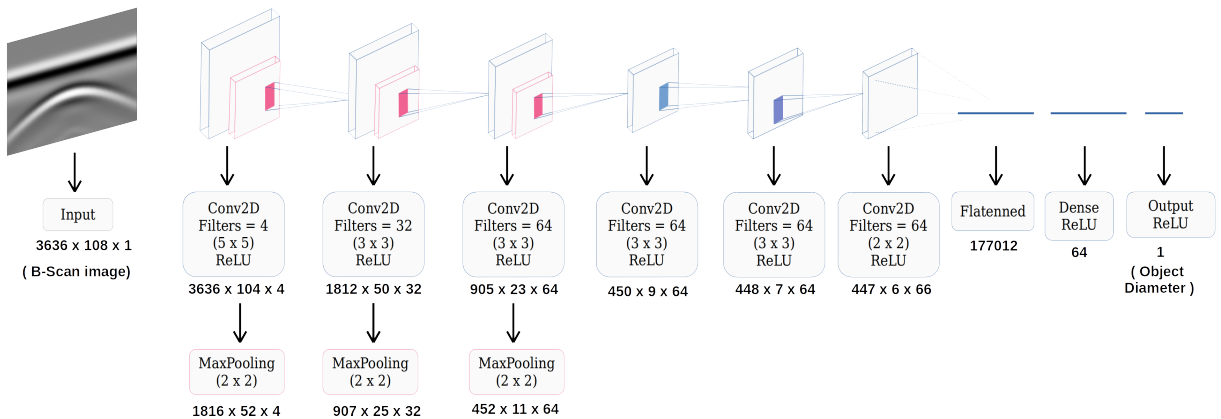


Figure 7: Architecture of the proposed CNN

6. Results and Discussion

It is observed that the model with Scharr edge detector followed by CNN model with 5 hidden layers performs the best with a validation loss of 6.94 (MAPE). However, Sobel, Laplacian, and Canny edge detectors did not give favourable results with the current dataset. Best MAPE (validation) for the CNN model with Sobel edge detector is 30.24 (7 hidden layers), Laplacian edge detector is 29.25 (6 hidden layers), and Canny edge detector is 18.51 (1 hidden layer).

Since the CNN model with Scharr edge detector shows the best validation loss, its model is saved in HDF5 format to evaluate its performance on other datasets. For a final performance evaluation of the model, another set of 100 B-Scans is created using gprMax. The MAPE of the model for this new and unseen dataset is found to be 7.49. Table 4 shows the performance of the model on the new dataset.

A violin plot showing the distribution of predicted values and their probability density is shown in Figure 8. It divides the data points (predicted values) into four equal portions (quartiles). The white dot in the middle denotes the median value and the thick black bar denotes the interquartile range. The thin black line extending on both sides represents the upper and lower adjacent values in the predicted data. The kernel density estimation on both sides of the black line shows the distribution of the data.

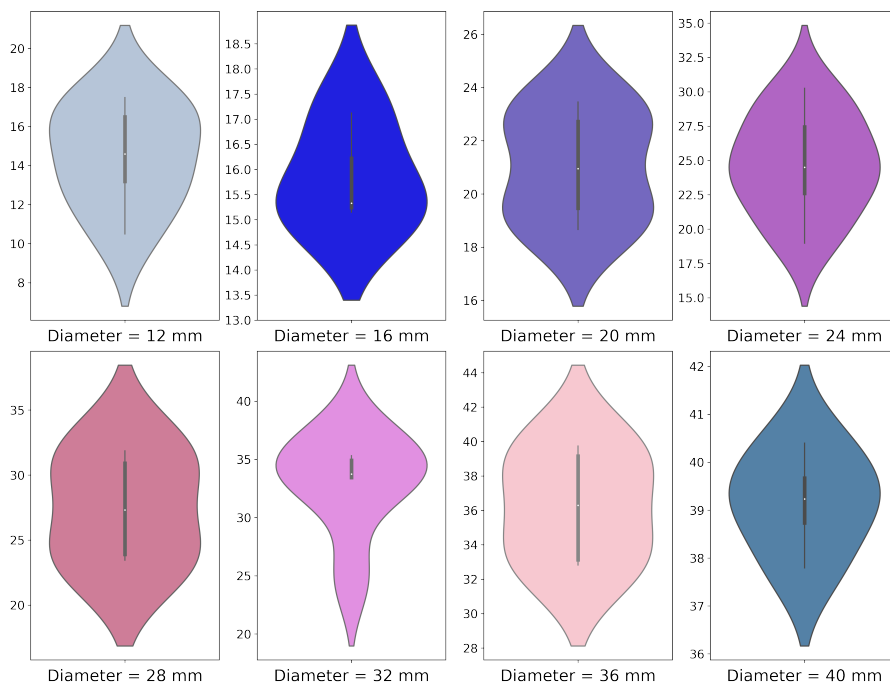
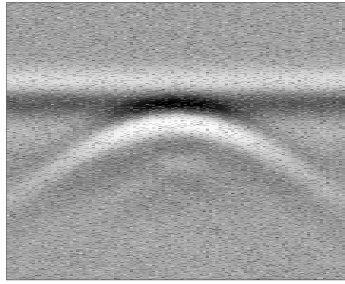


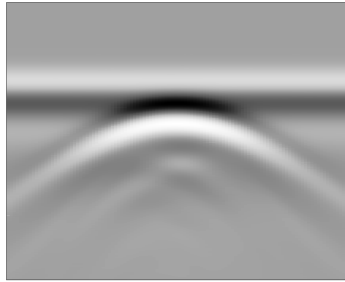
Figure 8: A violin plot showing the distribution of predicted values for various diameter values

In Figure 8, the plot for diameter 24 mm shows that the median value of the predicted values is 24.51 mm. For an actual value of 24 mm, the highest predicted value of 30.25 mm and the lowest predicted value of 23.93 mm create the boundaries of the interval space which contains all the data points. The interquartile range (IQR) shows the middle 50% of the four quartiles. In this case, the IQR ranges from 22.5 mm (first quartile) to 27.5 mm (third quartile) which means the middle 50% of the predicted values lie between 22.5 mm to 27.5 mm. The kernel density estimation (KDE) plot shows the probability distribution of the predicted values on both the sides of the vertical black line. The KDE plot goes to a maxima at around 24.51 mm which signifies that the probability of finding a predicted value near to 24.51 mm is maximum. This is in rough agreement with the fact that the actual value is 24 mm.

A comparison of the model's performance shows that Scharr edge detector (SED) followed by a 6 convolution layer CNN model boosts the overall performance over the other methods used. The SED approach extracts sufficient signature information related to the size of the object to raise the system's performance and reduce its validation MAPE to 6.74 and a MAPE of 7.49 on new, unseen data.



B-Scan with 10% noise



B-Scan after median filter

Figure 9

A GPR B-Scan with noise and after a median filter is applied to it

Table 5

Performance of CNN model on noisy data

Sl. No.	Noise Level	Diameter of target object (mm)		Error (mm)
		True value	Predicted value	
1	01%	36	36.92	0.92
2	02%	14	14.30	0.30
3	03%	36	34.03	1.97
4	04%	18	17.75	0.25
5	04%	17	17.70	0.70
6	04%	26	24.58	1.42
7	05%	22	23.58	1.58
8	05%	38	34.43	3.57
9	05%	24	20.84	3.16
10	10%	18	17.35	0.65
11	10%	26	24.85	1.15
12	10%	16	17.02	1.02
13	10%	28	25.47	2.53
14	10%	36	32.60	3.40
15	15%	28	26.47	1.53
16	15%	30	27.66	2.34
17	15%	16	14.03	1.97
18	15%	14	12.23	1.77
19	15%	26	29.80	3.80
20	20%	22	20.82	1.18
21	20%	14	14.91	0.91
22	20%	36	39.37	3.37
23	20%	28	21.77	6.23
24	20%	14	10.40	3.60
25	25%	20	15.58	4.42
26	25%	14	10.80	3.20
27	25%	26	19.01	6.99

6.1. Testing on noisy data

GPR images are seldom noise-free and ideal hyperbolic reflections are not obtained in most cases. Hence, it is imperative to test any feature extraction model on noisy data for proper evaluation of its performance in realistic scenarios.

For this, a new dataset is generated by adding noise randomly. Random number of pixels are selected for each GPR B-Scan and replaced with white and black pixels. The total number of pixels in each B-Scan being 3,92,688 (108 × 3636), the range of the randomly selected pixels is randomly varied from 0.3% (1400) to 25% (98000). Before applying the proposed model for diameter prediction, the image is first passed through a median filter which is a widely used non-linear filter to remove noise in GPR images. A Scharr filter is then applied on the image, after which the CNN model is used for diameter prediction. Figure 9 shows a GPR B-Scan with 10% noise added to it. Below it is the noise-free image after a median filter is used.

Several images containing various amounts of noise are evaluated using the proposed CNN model. Table 5 shows the performance of the model for different noise levels in the input image. As seen from the table, the model is able to predict the target's diameter with low error rates at low noise levels. However, as the noise level increases beyond 20%, it is seen that model's performance degrades with error rates of 22.26% at 20% noise level and upto 26.88% at 25% noise level.

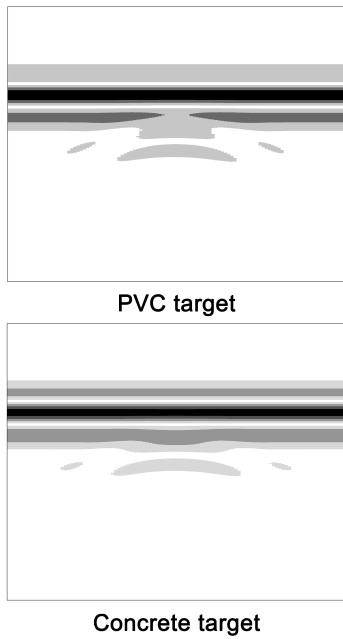


Figure 10
A GPR B-Scan with targets made of PVC and concrete

Table 6
Performance of CNN model for non-metallic objects

Sl. No.	Object Material	Diameter of target object (mm)		Error (mm)
		True values	Predicted values	
1	PVC	34	32.90	1.10
2	Concrete	18	19.28	1.28
3	Concrete	32	30.12	1.88
4	PVC	16	14.91	1.09
5	Concrete	26	28.39	2.39
6	PVC	20	21.83	1.83
7	PVC	30	31.36	1.36
8	PVC	10	9.79	0.21
9	Concrete	18	19.38	1.38
10	PVC	22	20.12	1.88
11	Concrete	34	32.11	1.89
12	Concrete	26	23.72	2.28
13	Concrete	20	18.44	1.56
14	Concrete	24	24.29	0.29
15	PVC	16	16.54	0.54
16	PVC	16	14.76	1.24
17	Concrete	38	34.59	3.41
18	Concrete	28	25.87	2.13
19	Concrete	28	25.41	2.59
20	Concrete	30	27.29	2.71

6.2. Testing on non-metallic buried objects

A CNN model for detecting and predicting the size of buried objects will not be considered for practical application if it can be applied to metallic targets only. Hence, its performance evaluation is needed for non-metallic targets as well. For this purpose, a dataset consisting of PVC ($\sigma = 0.001$ Siemens/metre, $\epsilon_r = 4$) and concrete ($\sigma = 0.01$ Siemens/metre, $\epsilon_r = 7.3$) is generated. Figure 10 shows a GPR B-Scans after a application of Scharr filter for PVC and concrete targets.

On evaluation of the CNN model of this dataset, it is seen that it can predict the target's size with very low error rates, irrespective of the target material. The dataset is generated using the methodology as previously described in section 2. Table 6 lists the values predicted by the model for different target materials having different sizes.

6.3. Testing on multiple buried objects

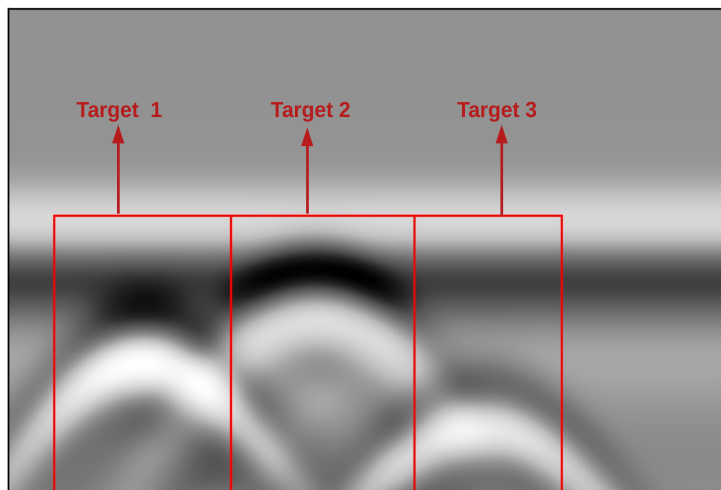


Figure 11: A GPR B-Scan showing 3 buried objects

To test the efficacy of the model when multiple buried objects are present, few B-Scans are generated containing

upto 3 objects of different sizes and at different depths. All the objects are considered to be cylindrical in shape. One such B-Scan image is shown in Figure 11 where multiple hyperbolas with overlapping edges can be seen. It contains 3 metallic cylinders at depths of diameters 30 mm, 20 mm, and 40 mm from left-to-right.

In the case of multiple target objects, the hyperbolas will have to be isolated from the image before the model can be used for estimation of target size. The hyperbolas can be isolated using template matching, which is a commonly used algorithm in image processing. They can also be extracted manually by cropping the image. Next, each image containing the individual hyperbola is reshaped to have the same size as required by the CNN model, in this case 108×3636 . This is done by padding around the edges of each image with linearly varying values from the pixels at the edge of the image to the 3636th / 108th pixel.

As shown in Figure 12, a Scharr filter is finally applied to each image after which the CNN model is used to estimate the size of the targets. The model shows decent performance in estimating the size of the targets. It predicted a size of 34.46 mm for the object with 34 mm actual diameter, 22.86 mm for 20 mm (actual), and 44.15 mm for an actual diameter of 40 mm. However, those performances in which each image recorded a single hyperbola were better than this result.

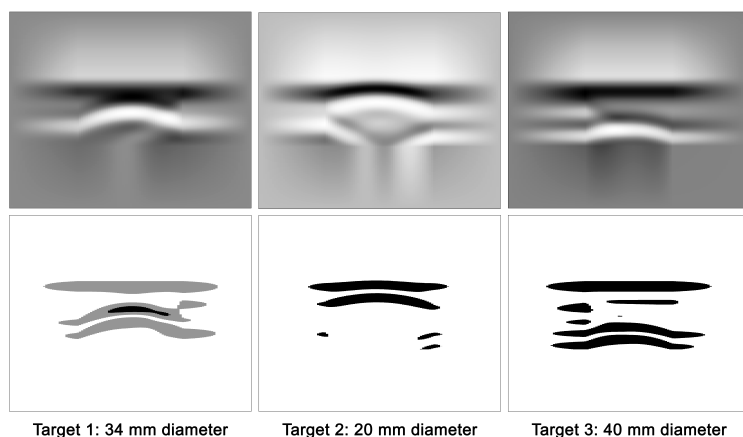


Figure 12: The extracted hyperbolas before and after application of Scharr filter

7. Conclusion

A novel approach of using CNN for predicting size of buried objects from GPR B-Scans was presented. The proposed CNN architecture extracts high level signatures in the first stage of feature extraction and learns additional low level features when the input data passes through the neural network to finally make an estimation of the required parameter. A comparison with other techniques is shown in Table 7 from which it can be concluded that the application of CNN in this problem has demonstrated a good performance in predicting size of buried objects. Moreover, it can be used in real time applications in GPR data interpretation since such a model can be implemented on embedded devices like ARM, FPGA, etc.

Future work of this approach will concentrate on experimentally verifying the model with actual data taken from an FPGA-based GPR system being developed in-house as well as from commercial GPR systems. The present model was trained with an object which is a metallic cylinder made of aluminium. Even though the model was tested on PVC and concrete objects with good results, its performance can still be improved by training it using greater number of datasets containing objects of a variety of materials. Different types of materials will have different dielectric properties and hence will generate different scattering and dispersion characteristics. Similarly, training the model for different shapes such as rectangular and triangular forms give make it more robust. It remains to be seen how these will affect the robustness of the proposed approach.

Acknowledgements

The authors would also like to thank Dr. Manoj Kumar Phukan (Sr. Scientist, Geo Sciences and Technology Division, CSIR-NEIST, Jorhat) who gave detailed insights on interpretation of GPR data and Ms. Sharmistha Mazumdar for her help in visualising the results. Finally, the authors appreciate the help of Mr. Arnob Doloi for his inputs during

Table 7

Comparison of present work with past studies

Sl. No.	Author	Technique used	Application	Real time / Offline
1	Chaudhuri and Samanta (1991)	Hyperbolic feature extraction using Moment of inertia optimisation	Extraction of hyperbolic features	Offline
2	Delbo et al. (2000)	Hyperbolic feature extraction using Fuzzy shell clustering	Extraction of hyperbolic features	Offline
3	Ko et al. (2012)	PCA and Fourier Transform	Detection of landmines	Offline
4	Wahab et al. (2013)	Hyperbola fitting technique	Estimation of radius of pipes and cables	Offline
5	Maas and Schmalzl (2013)	Canny operator for edge detection and Hough transform for hyperbola fitting	Automatically localize reflection hyperbola	Real time
6	Besaw and Stimac (2015)	ANN based Deep Belief Network (DBN)	Detection of explosive hazards	Offline (in situ)
7	Mertens et al. (2015)	Hyperbolic feature extraction using Canny operator	Automated detection of reflection hyperbola	Offline
8	Dou et al. (2016)	Canny operator for edge detection and Hough transform for hyperbola fitting	Automatically localize reflection hyperbola	Real time
9	Li et al. (2016)	Hyperbolic feature extraction using Sobel operator followed by Hough Transform	Automatic recognition of tree roots	Offline
10	Lameri et al. (2017)	Deep Convolutional Neural Network	Detection of landmines	Real Time
11	Todkar et al. (2017)	Machine learning using Support Vector Machines (SVM)	Detection of debondings	Offline (in situ)
12	Kafedziski et al. (2018)	Faster Region Convolutional Neural Network (Faster R-CNN)	Detection and classification of landmines	Real Time
13	Sonoda and Kimoto (2018)	Deep CNN	Object identification	Real Time
14	Kang et al. (2020)	Basis pursuit-based background filtering algorithm followed by deep CNN	Underground cavity detection	Real Time
15	Bugarinović et al. (2020)	Hyperbolic feature extraction using Canny operator	Automated detection of hyperbolic features	Offline
16	Giannakis et al. (2021)	Autoencoder as a pre processing step followed by ANN based regressor	Estimate the diameter of reinforcement bars in concrete	Real Time
17	Present work	Hyperbolic feature detection followed by deep CNN	Estimation of radius/ diameter of buried objects	Real time

database creation from time to time. The authors would also like to thank Ms. Sweta Kumar Gaur for her insightful suggestions in linguistic polish of the manuscript.

Data Availability

All the data, models, or codes generated during this study are available from the corresponding author upon request.

CRedit authorship contribution statement

Nairit Barkataki: Conceptualization, Methodology, Software, Writing - Original Draft. **Banty Tiru:** Supervision, Software, Writing - Review & Editing. **Utpal Sarma:** Supervision, Resources, Writing - Review & Editing.

References

- Adhikari, R., Moselhi, O., Bagchi, A., 2014. Image-based retrieval of concrete crack properties for bridge inspection. *Automation in construction* 39, 180–194.
- Al-Nuaimy, W., Huang, Y., Nakhkash, M., Fang, M., Nguyen, V., Eriksen, A., 2000. Automatic detection of buried utilities and solid objects with gpr using neural networks and pattern recognition. *Journal of applied Geophysics* 43, 157–165.
- Bergholm, F., 1987. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 726–741. doi:10.1109/TPAMI.1987.4767980.
- Besaw, L.E., Stimac, P.J., 2015. Deep convolutional neural networks for classifying gpr b-scans, in: *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, International Society for Optics and Photonics. p. 945413.
- Botev, A., Lever, G., Barber, D., 2017. Nesterov's accelerated gradient and momentum as approximations to regularised update descent, in: *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE. pp. 1899–1903.
- Bugarinović, Ž., Pajewski, L., Ristić, A., Vrtunski, M., Govedarica, M., Borisov, M., 2020. On the introduction of canny operator in an advanced imaging algorithm for real-time detection of hyperbolas in ground-penetrating radar data. *Electronics* 9, 541.

- Chaudhuri, B.B., Samanta, G., 1991. Elliptic fit of objects in two and three dimensions by moment of inertia optimization. *Pattern Recognition Letters* 12, 1–7.
- Chen, G., Jiang, Z., Kamruzzaman, M., 2020. Radar remote sensing image retrieval algorithm based on improved sobel operator. *Journal of Visual Communication and Image Representation* 71, 102720.
- Chen, H., Cohn, A.G., 2010. Probabilistic robust hyperbola mixture model for interpreting ground penetrating radar data, in: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, IEEE. pp. 1–8.
- Chen, S., Wang, H., Xu, F., Jin, Y.Q., 2016. Target classification using the deep convolutional networks for sar images. *IEEE Transactions on Geoscience and Remote Sensing* 54, 4806–4817.
- Cheng, J.C., Wang, M., 2018. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Automation in Construction* 95, 155–171.
- Delbo, S., Gamba, P., Roccato, D., 2000. A fuzzy shell clustering approach to recognize hyperbolic signatures in subsurface radar images. *IEEE Transactions on Geoscience and Remote Sensing* 38, 1447–1451.
- Dell’Acqua, A., Sarti, A., Tubaro, S., Zanzi, L., 2004. Detection of linear objects in gpr data. *Signal Processing* 84, 785–799.
- Di, H., Gao, D., 2014. Gray-level transformation and canny edge detection for 3d seismic discontinuity enhancement. *Computers & Geosciences* 72, 192–200.
- Ding, J., Chen, B., Liu, H., Huang, M., 2016. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters* 13, 364–368.
- Dou, Q., Wei, L., Magee, D.R., Cohn, A.G., 2016. Real-time hyperbola recognition and fitting in gpr data. *IEEE Transactions on Geoscience and Remote Sensing* 55, 51–62.
- Frigui, H., Gader, P., 2008. Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k -nearest neighbor classifier. *IEEE Transactions on Fuzzy Systems* 17, 185–199.
- Gamba, P., Lossani, S., 2000. Neural detection of pipe signatures in ground penetrating radar images. *IEEE Transactions on Geoscience and Remote Sensing* 38, 790–797.
- Giannakis, I., Giannopoulos, A., Warren, C., 2021. A machine learning scheme for estimating the diameter of reinforcing bars using ground penetrating radar. *IEEE Geoscience and Remote Sensing Letters* 18, 461–465.
- Guo, K., Sui, L., Qiu, J., Yu, J., Wang, J., Yao, S., Han, S., Wang, Y., Yang, H., 2017. Angel-eye: A complete design flow for mapping cnn onto embedded fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 35–47.
- Jacob, R.W., Urban, T., 2016. Ground-penetrating radar velocity determination and precision estimates using common-midpoint (cmp) collection with hand-picking, semblance analysis and cross-correlation analysis: a case study and tutorial for archaeologists. *Archaeometry* 58, 987–1002.
- Jiang, J., Zhang, Z., Dong, Q., Ni, F., 2018. Characterization and identification of asphalt mixtures based on convolutional neural network methods using x-ray scanning images. *Construction and Building Materials* 174, 72–80.
- Kafedziski, V., Pecov, S., Tanevski, D., 2018. Detection and classification of land mines from ground penetrating radar data using faster r-cnn, in: *2018 26th Telecommunications Forum (TELFOR)*, IEEE. pp. 1–4.
- Kang, M.S., Kim, N., Lee, J.J., An, Y.K., 2020. Deep learning-based automated underground cavity detection using three-dimensional ground penetrating radar. *Structural Health Monitoring* 19, 173–185.
- Ko, K.H., Jang, G., Park, K., Kim, K., 2012. Gpr-based landmine detection and identification using multiple features. *International journal of antennas and propagation* 2012.
- Koch, C., Georgieva, K., Kasiredy, V., Akinci, B., Fieguth, P., 2015. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics* 29, 196–210.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems*, Neural Information Processing Systems Foundation. pp. 1097–1105.
- Lameri, S., Lombardi, F., Bestagini, P., Lualdi, M., Tubaro, S., 2017. Landmine detection from gpr data using convolutional neural networks, in: *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE. pp. 508–512.
- Li, W., Cui, X., Guo, L., Chen, J., Chen, X., Cao, X., 2016. Tree root automatic recognition in ground penetrating radar profiles based on randomized hough transform. *Remote Sensing* 8, 430.
- Liu, C., Tang, C.S., Shi, B., Suo, W.B., 2013. Automatic quantification of crack patterns by image processing. *Computers & Geosciences* 57, 77–80.
- Liu, H., Lin, C., Cui, J., Fan, L., Xie, X., Spencer, B.F., 2020. Detection and localization of rebar in concrete by deep learning using ground penetrating radar. *Automation in Construction* 118, 103279.
- Liu, X., Cui, X., Guo, L., Chen, J., Li, W., Yang, D., Cao, X., Chen, X., Liu, Q., Lin, H., 2019. Non-invasive estimation of root zone soil moisture from coarse root reflections in ground-penetrating radar images. *Plant and Soil* 436, 623–639.
- Liu, Y., Durlofsky, L.J., 2020. 3d cnn-pca: A deep-learning-based parameterization for complex geomodels. *Computers & Geosciences* , 104676.
- Lu, Q., Pu, J., Liu, Z., 2014. Feature extraction and automatic material classification of underground objects from ground penetrating radar data. *Journal of Electrical and Computer Engineering* 2014.
- Maas, C., Schmalzl, J., 2013. Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar. *Computers & geosciences* 58, 116–125.
- Mertens, L., Persico, R., Matera, L., Lambot, S., 2015. Automated detection of reflection hyperbolas in complex gpr images with no a priori knowledge on the medium. *IEEE Transactions on Geoscience and Remote Sensing* 54, 580–596.
- Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E., 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, 1.
- Nguyen, D.T., Nguyen, T.N., Kim, H., Lee, H.J., 2019. A high-throughput and power-efficient fpga implementation of yolo cnn for object detection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 1861–1873.
- Pasolli, E., Melgani, F., Donelli, M., 2009. Gaussian process approach to buried object size estimation in gpr images. *IEEE Geoscience and Remote Sensing Letters* 7, 141–145.

- Ristić, A., Bugarinović, Ž., Vrtunski, M., Govedarica, M., 2017. Point coordinates extraction from localized hyperbolic reflections in gpr data. *Journal of Applied Geophysics* 144, 1–17.
- Shaw, M., Millard, S., Molyneaux, T., Taylor, M., Bungey, J., 2005. Location of steel reinforcement in concrete using ground penetrating radar and neural networks. *Ndt & E International* 38, 203–212.
- Shihab, S., Al-Nuaimy, W., Eriksen, A., 2002. Image processing and neural network techniques for automatic detection and interpretation of ground penetrating radar data. *Proceedings of the 6th International Multi-Conference on Circuits, Systems, Communications and Computers, Cancun, Mexico*, 12–16.
- Smitha, N., Singh, V., 2020. Target detection using supervised machine learning algorithms for gpr data. *Sensing and Imaging* 21, 1–15.
- Sonoda, J., Kimoto, T., 2018. Object identification from gpr images by deep learning, in: *2018 Asia-Pacific Microwave Conference (APMC)*, IEEE. pp. 1298–1300.
- Syambas, N.R., 2012. An approach for predicting the shape and size of a buried basic object on surface ground penetrating radar system. *International Journal of Antennas and Propagation* 2012, 1–13.
- Todkar, S.S., Le Bastard, C., Ihamouten, A., Baltazart, V., Dérobert, X., Fauchard, C., Guilbert, D., Bosc, F., 2017. Detection of debondings with ground penetrating radar using a machine learning method, in: *2017 9th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, IEEE. pp. 1–6.
- Tong, Z., Gao, J., Zhang, H., 2018. Innovative method for recognizing subgrade defects based on a convolutional neural network. *Construction and Building Materials* 169, 69–82.
- Torrione, P.A., Morton, K.D., Sakaguchi, R., Collins, L.M., 2013. Histograms of oriented gradients for landmine detection in ground-penetrating radar data. *IEEE transactions on geoscience and remote sensing* 52, 1539–1550.
- Von Hippel, A.R., 1954. *Dielectric Materials and Applications; papers by twenty-two contributors.* published jointly by the Technology Press of M.I.T. and Wiley.
- Wahab, W.A., Jaafar, J., Yassin, I.M., Ibrahim, M.R., 2013. Interpretation of ground penetrating radar (gpr) image for detecting and estimating buried pipes and cables, in: *2013 IEEE International Conference on Control System, Computing and Engineering*, IEEE. pp. 361–364.
- Warren, C., Giannopoulos, A., Giannakis, I., 2016. gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar. *Computer Physics Communications* 209, 163–170.
- Warren, C., Giannopoulos, A., Gray, A., Giannakis, I., Patterson, A., Wetter, L., Hamrah, A., 2019. A cuda-based gpu engine for gprmax: Open source fdtd electromagnetic simulation software. *Computer Physics Communications* 237, 208–218.
- Windsor, C., Capineri, L., Falorni, P., Matucci, S., Borgioli, G., 2005. The estimation of buried pipe diameters using ground penetrating radar. *Insight-Non-Destructive Testing and Condition Monitoring* 47, 394–399.
- Wiwatrojanagul, P., Sahamitmongkol, R., Tangtermsirikul, S., Khamsemanan, N., 2017. A new method to determine locations of rebars and estimate cover thickness of rc structures using gpr data. *Construction and Building Materials* 140, 257–273.
- Zakeri, H., Nejad, F.M., Fahimifar, A., 2017. Image based techniques for crack detection, classification and quantification in asphalt pavement: a review. *Archives of Computational Methods in Engineering* 24, 935–977.
- Zhang, C., Elaksher, A., 2012. An unmanned aerial vehicle-based imaging system for 3d measurement of unpaved road surface distresses 1. *Computer-Aided Civil and Infrastructure Engineering* 27, 118–129.
- Zhang, Y., Huston, D., Xia, T., 2016. Underground object characterization based on neural networks for ground penetrating radar data, in: Yu, T., Gyekenyesi, A.L., Shull, P.J., Wu, H.F. (Eds.), *Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, and Civil Infrastructure 2016*, International Society for Optics and Photonics. SPIE. pp. 10 – 18. doi:10.1117/12.2219345.
- Zhao, S., Al-Qadi, I.L., 2016. Development of an analytic approach utilizing the extended common midpoint method to estimate asphalt pavement thickness with 3-d ground-penetrating radar. *NDT & E International* 78, 29–36.