

Enhancing Accuracy for Metal Target Detection Using CNN-GP Algorithm

Xiaofen Wang¹, Xiaotong Zhang^{1,2,*}, Yadong Wan^{1,*}, and Peng Wang³

¹ Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China

² Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, Beijing, China

³ China Unicom Smart City Research Institute, Beijing, China

wxf@xs.ustb.edu.cn, zxt@ies.ustb.edu.cn, wyd@ustb.edu.cn,
wangpeng.mic1@gmail.com

Abstract. Inversion based on electromagnetic induction (EMI) is an important method for detection of underground metal targets in fields such as archaeology and geological exploration. However, traditional inversion algorithms grounded in the framework of least squares suffer from long iteration times, susceptibility to local optima, and dependence on initial values. To address these challenges and enhance the detection of underground metal target detection, this paper proposes an innovative CNN-GP algorithm based on Convolutional Neural Network (CNN) and Gaussian Process (GP). Our proposed algorithm initiates by extracting discriminative features based on CNN, followed by dimensionality reduction through a multilayer perceptron (MLP) to map the extracted features into low-dimensional vectors, and estimating the position of metal targets through GP algorithm. To refine the accuracy of the CNN-GP algorithm, this paper uses grid and Bayesian search algorithms for network optimization. Results demonstrate that the Bayesian search algorithm expeditiously identifies an optimal set of hyperparameters, yielding inversion performance compared with grid search algorithm. Comparative analyses of inversion efficacy between CNN, GP, MLP, and CNN-GP algorithms pre- and post-optimization reveal CNN-GP as the optimal performer, with inversion errors of 0.5cm, 0.5cm, and 2.4cm along the x , y , and z direction, respectively.

Keywords: underground metal target detection, electromagnetic induction, CNN, hyperparameter optimization, transfer learning.

1 Introduction

Underground metal target detection is an important branch of geophysical inversion, which finds widespread applications in fields such as resource exploration [1], archaeology [2], detection of underground infrastructure [3], and unexploded ordnance detection [4]. Detecting metal targets buried underground provide detailed physical information about the metal, such as its location, orientation, and resistivity. This process of

obtaining physical parameters of underground metal targets is called inversion. With the development of computer technology, there are higher requirements for detection performance, and faster response times are necessary when accurately inverting the physical properties of underground metal targets.

In general, inversion algorithms for underground metal targets include model-based inversion algorithms and data-driven inversion algorithms. Model-based inversion algorithms operate on the principle of estimating the physical properties of underground metal targets by solving for the model parameters that minimize a least squares objective function. The effectiveness of model-based inversion algorithms depends on the rational design of the objective function. On the other hand, data-driven inversion algorithms predominantly use artificial intelligence techniques such as machine learning and deep learning. These methods establish mappings between response data and the physical properties to be inverted. Typically, these algorithms take either raw or feature-extracted response data as input and the corresponding true physical properties as labels for model training. Once trained, these models can then be used to estimate the physical properties of metal targets.

Deep learning, as a popular machine learning technique, has been widely applied in various domains. Using deep learning algorithms, applications in seismic data inversion [5], underground cavity detection [6], and other areas have yielded promising results. This highlights the potential of deep learning algorithms to address the nonlinear and non-unique inversion problems in geophysical domain. Traditional inversion algorithms based on least squares suffer from problems such as dependence on initial values, long iteration times, and susceptibility to local optima. Furthermore, in the study of Wang et al. [7], the objective function based on the least squares optimization algorithm is designed according to the forward physical model used to generate simulation data, which may lead to potential inverse crime [8], where inversion performs well on simulation data but poorly on real collected data. Therefore, the use of deep learning-based methods for inversion provides a viable approach to overcome the limitations of traditional inversion algorithms. In addition, CNNs are a commonly used method when processing data with spatial characteristics. Therefore, exploring the potential of CNNs in inversion for underground metal target holds significant research value and significance.

Deep learning algorithms are widely used in the electromagnetic field. Lei et al. [3] utilized a CNN-LSTM hybrid model to detect and estimate the diameter of cylindrical metal targets. The CNN processed two-dimensional magnetic map data from GPR B-SCAN, while the LSTM handled one-dimensional temporal data from GPR A-SCAN. Results from both layers were integrated to derive the final inversion outcome, which achieves over 90% accuracy for simulated and real data. Puzyrev et al. [9] employed CNNs for time-domain inversion to predict resistivity in underwater and underground scenarios and demonstrated the superior performance and efficiency of CNNs over the LM algorithm. By training a fully convolutional network on simulated datasets and validating on real-world scenarios, Davood et al. [10] proposed a novel deep CNN-based method for inversion of EMI data. Results addressed computational complexity, and nonlinearity, and achieved quick estimations in milliseconds, beneficial for large-scale prediction tasks. Puzyrev et al. [11] proposed a fully convolutional deep neural

network for predicting underground resistivity, utilizing simulation data for model training. Experimental examples across different scenarios and noise levels confirmed the feasibility of deep learning inversion. Aleardi et al. [12] used a convolution-based approach to address resistivity tomography inversion, which compressed data via discrete cosine transform and trained on transfer learning methods to facilitate real-time estimation of underground resistivity.

Therefore, this paper focuses on the application of deep CNNs in the field of underground metal target detection. Taking underground metal target detection as the background, cylindrical metal objects are considered as targets for underground detection. A CNN structure is designed, and the model is trained using simulated data generated by a dipole model-based forward. The trained model facilitates the inversion of underground metal target properties. The current model cannot be directly extrapolated to real-world datasets due to potential disparities between simulated and real data. Considering that both tasks involve feature extraction from magnetic maps, this paper uses model-based transfer learning methods to transfer the model trained on simulated data to the prediction task of real data.

This paper provides the following contribution:

- The CNN-GP algorithm is proposed to apply CNN to the inversion problem of underground metal target detection, to extract feature data directly in the magnetic map and find the mapping relationship between magnetic field data and metal target parameters. CNN-GP algorithm is an efficient tool for dealing with data inversion in geophysics, which solves the mismatch between high-precision instrument-acquired data and lightweight data processing methods.
- This paper optimizes the hyperparameters of the CNN-GP algorithm using the grid and the Bayesian search algorithm to improve the model learning efficiency and prediction accuracy.
- The model-based transfer learning is used to solve the problem of difficult model reproduction due to the difference between simulated and real data. This paper keeps the feature extraction part related to CNN and re-learn the regression part of the model on real data to effectively improve the learning ability of the model.

2 Related Work

2.1 Metal Target Detection

The theoretical core of frequency-domain electromagnetic detection methods lies in Faraday's law of electromagnetic induction, which detects changes in the magnetic field of targets to achieve detection. Currently, electromagnetic detection devices include the EM series developed by Geonics [13], the GEM series developed by Geophex [14], the NEMFIS series developed in Russia [15], GCM [16], CEM [17], and others. These devices have similar hardware architectures, consisting mainly transmitters, receivers, and control units.

The process of generating secondary field data can be described as:

$$I_p(t) \rightarrow \mathbf{B}_p(r,t) \rightarrow \mathbf{J}(t) \rightarrow \mathbf{B}_p(r,t) + \mathbf{B}_s(r,t) \rightarrow \mathbf{V}_s(t) \quad (1)$$

The transmitter drives the transmitting coil to emit a changing current I_p , thereby generating low-frequency electromagnetic waves to the ground, known as the primary field B_p . Underground metal targets, exposed to the primary field generate eddy currents internally, which in turn generate a magnetic field known as the secondary field B_s . The rapidly decaying secondary field is finally received by the receiver and transmitted to the control unit for further processing, resulting in the final response signal V_s .

In the field of underground metal target detection, the most used data acquisition scheme is 2D parallel scanning, as shown in Fig. 1. During data collection through a portable handheld device, a square area is selected on the ground, starting from one corner of the square area, and data is collected at certain sampling spaces.

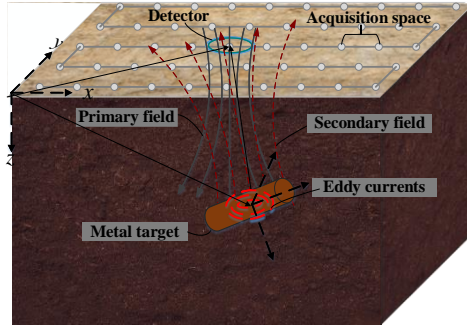


Fig. 1. Process of underground metal target detection

Upon obtaining response data, inversion methods can be used to estimate the position information and physical properties of underground metal targets. For model-based inversion algorithms, initial estimates are typically initialized before inversion based on prior information. By introducing a dipole model into the objective function, the difference between the estimated and collected response data can be calculated through forward computation. This difference is used to iteratively update the estimated values until a predefined number of iterations is reached or the difference falls below a specific threshold, indicating the completion of the inversion. In contrast, data-driven inversion algorithms use collected response data directly as the input data to train the model and further estimate the relevant properties of underground metal targets.

2.2 Convolutional Neural Network

Convolutional Layer. The convolutional layer is one of the most important components of a CNN, composed of convolutional kernels, where each kernel is a small matrix. When dealing with images like color images that require multiple channels, the image is represented as a multi-dimensional matrix, and convolutional operations are used to extract features such as edges, textures, and colors. Through feature extraction, the convolutional layer transforms raw pixels into higher-level feature, facilitating further tasks such as image classification and recognition.

During feature extraction, the most important step is to perform convolution on each sub-matrix of the input matrix to obtain a new matrix representing the extracted features. The convolution is expressed as follows:

$$(s \times t)(n) = \int_{-\infty}^{\infty} s(\tau)t(n-\tau)d\tau \quad (2)$$

where s and t are two continuous functions and the convolution of the two is described in terms of an integral sum. In image processing, the convolution of two matrices can be expressed in terms of a two-dimensional discrete equation. Assuming that the size of the convolution kernel is $m \times n$, the convolution of the image sub-matrix and the convolution kernel can be represented by the following equation:

$$s(x, y) \times t(x, y) = \sum_{i=0}^{i=m} \sum_{j=0}^{j=n} s(i, j) \times t(x-i, y-j) \quad (3)$$

Pooling Layer. The pooling layer is used to perform dimensionality reduction on the input data through methods such as subsampling, average pooling, min pooling, or max pooling. This helps to remove redundant information and reduces the computational complexity of the model. During the operations of pooling, it is necessary to set the size and stride of the pooling kernel. Pooling filters out redundant information from the image while preserving important features, thus improving the performance of the model for better results. However, improper settings can also result in relatively important features being filtered out, which will have a negative impact on the performance of the model.

Fully Connected Layer. The fully connected (FC) layer is usually located at the end of the entire CNN structure and follows the basic approach of a traditional MLP. Its input comes from the feature vector created by the Flatten layer, which flattens the feature matrix from the last layer of the CNN into a vector. Then, the fully connected layer performs the inversion.

The loss function is critical in all supervised learning algorithms and is used to measure the difference between the predicted and actual output of the model. Gradient-based methods such as Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), and Mini-Batch Gradient Descent (MGD) are commonly used to train CNNs. These methods continuously update the network parameters overall training epochs to minimize the value of the loss function. This process occurs overall training epochs until the network searches for the optimal solution to minimize the error.

2.3 Hyperparameter Optimization

It is essential to improve the performance of the network by selecting appropriate hyperparameters, such as optimizer, batch size, and learning rate (LR). Currently, the most used optimizers include variants of SGD with momentum, such as AdaGrad [18], RMSprop [19], and Adam [20]. On the other hand, hyper-parameters related to the model mainly include the structure of the neural network, such as the number and width of hidden layers.

Learning Rate. To improve the accuracy of the model, it is generally necessary to manually adjust the LR to balance training time and convergence speed [21]. In addition, adaptive learning rate adjustment adjusts automatically based on the performance or structure of the model and has been supported by relevant learning algorithms [22].

Optimizer-related Hyperparameters. In practice, Adam is recommended as the default optimization algorithm for training deep learning models. It has more hyperparameters than other optimizers but works well with minimal adjustment of parameters other than the LR. The Adam sets the LR to start from 0.001 and gradually decrease, with a default momentum parameter of 0.9, an RMSprop parameter of 0.999, and a smoothing term of $1e-8$. The default values of these hyperparameters are suitable for most problems.

Model-related Hyperparameters. The number of hidden layers is a key hyperparameter related to the network structure, which directly affects the final performance of the trained network model [23]. Generally, increasing the number of hidden layers in a deep learning network would improve the performance of the neural network. However, care must be taken to avoid potential problems such as overfitting. In addition to the number of hidden layers, careful consideration must be given to the number of neurons within each layer.

The dropout rate is a hyperparameter that controls the rate at which neurons are randomly dropped out during training. A dropout rate that is too high may overly simplify the model, while one that is too low may not have a significant effect.

Activation functions in deep learning introduce non-linear properties allowing neural networks to handle complex features. Activation functions must be differentiable to compute weight gradients and improve training efficiency.

2.4 Model-based Deep Transfer Learning

One unavoidable challenge in the electromagnetic inversion problem of underground metal targets addressed in this paper is the availability of training data. Real-world data is much more difficult to obtain than simulated data. One approach to address this problem is to transfer the knowledge learned from simulated data to fit the inversion problem of real detection scenarios. This process, known as transfer learning, involves transferring knowledge learned from the source domain to the target domain. In this way, the training and testing data do not need to be completely independently and identically distributed, thereby addressing the shortage of training data and significantly reducing the learning time in the target domain.

The core idea of model-based transfer learning is to incorporate the well-trained network model from the source domain into the target domain network, allowing the knowledge learned from the source domain to be utilized for tasks in the target domain. For some generalized features, such as those extracted from input data using convolution, more accurate information can be provided for tasks in the target domain, thereby accelerating model training and improving performance. Pre-trained networks in the source domain have learned common features and patterns, thereby avoiding building

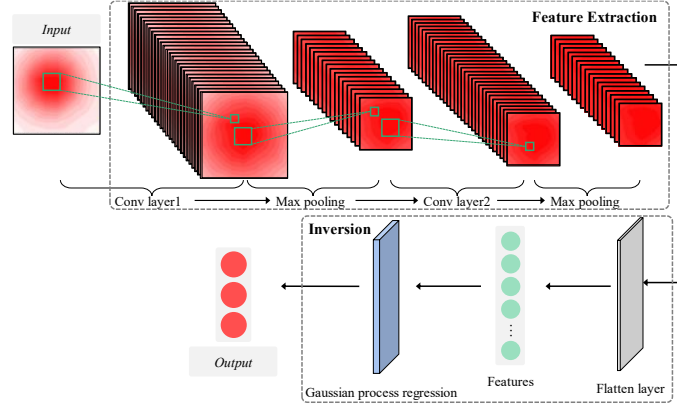


Fig. 2. Proposed CNN-GP algorithm.

a model from scratch.

In the study of George et al. [24], knowledge obtained from object recognition tasks was transferred to fault classification tasks in gravitational wave signal detectors using DNNs as feature extractors, resulting in promising outcomes. In the recognition task, to maximize the effectiveness of models trained with small samples, Oquab et al. [25] used partially trained models from the ImageNet dataset to obtain intermediate results for other image datasets. Furthermore, in language-related tasks, to increase efficiency by transferring features between different contexts and languages, Huang et al. [26] divided the network into language-related and non-language-related transformers and classifiers.

3 Method

3.1 Model Construction

Network Architecture The CNN-GP algorithm, as shown in Fig. 2, uses CNN to extract features from the secondary field magnetic map, followed by the GP regression and the MLP to further invert the extracted features to obtain the parameters of underground metal targets.

The proposed CNN-GP algorithm consists primarily of CNN and GP processing modules. The CNN module comprises two convolution layers and two max-pooling layers for extracting magnetic map features. Each convolution layer has a kernel size of 3×3 , with a stride of 1, and the number of kernels is 32 and 64, respectively. A max-pooling layer is added after each convolutional layer. The pooling window size for each pooling layer is 2×2 , with a stride of 2. Additionally, the CNN applies padding to the edges to prevent loss of edge information.

Since the input data is an image, it needs to be converted to a one-dimensional vector through a flatten layer before entering the Gaussian process regression. Gaussian process regression is a Bayesian non-parametric kernel method that models function

directly, thereby generating non-parametric models. It models the mapping relationship between observed data and the parameters of metal targets by using a covariance function, also known as the kernel function. The effectiveness of GP regression depends on the choice of kernel function. The Maternal and WhiteKernel kernel functions are used for preliminary experiments prior to optimization. The number of optional resets of the optimizer is set to [0, 5, 10, 15], which will be further studied in the model optimization subsection. Before entering the MLP, a flattening layer is added to transform the multi-dimensional matrix into a one-dimensional vector. MLP contains two dense layers, which have 256 and 64 neurons respectively. Finally, the size of the output layer is consistent with the dimensionality of the parameters to be inverted.

The input data consists of magnetic maps, where the scale of the magnetic map is determined by the sampling space. A smaller sampling space results in higher dimensional magnetic maps. The magnetic map used in this paper is generated with a sampling space of 0.5m.

Loss Function Selection. The mean absolute error (MAE) and mean squared error (MSE) are the most used to train a model. MAE represents the average of the absolute differences between predicted values and observed values, which can be described by the following equations:

$$f_{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*| \quad (4)$$

MSE can be expressed as:

$$f_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2 \quad (5)$$

where y_i represents the observed value, y_i^* represents the predicted value, and n represents the number of data points in the training dataset. In this paper, MSE is employed as the loss function during model training, while MAE is adopted as the evaluation metric for predictions on the test dataset.

Optimization. Commonly used optimization algorithms include SGD and the Adam optimizer. When using the SGD optimizer for training, attention should be paid to convergence issues. Adam optimizer leverages first, and second-order moment estimates of gradients to dynamically adjust weights and biases, enabling more efficient network optimization. Additionally, Adam possesses advantages such as adaptivity and bias correction, automatically adjusting the LR during training to enhance the performance of the model. Therefore, Adam is chosen as the optimizer in this paper.

Activation Function. In this paper, the Leaky ReLU function is used as the activation function. The Leaky ReLU function differs from the ReLU function in that it scales down negative inputs proportionally, thus preventing the complete neglect of negative inputs and addressing the issue of Dying ReLU. Leaky ReLU function can be expressed as Eq. (6):

$$f(x) = \begin{cases} x, & x > 0 \\ mx, & x \leq 0 \end{cases} \quad (6)$$

where m represents a fixed parameter, which is typically assigned a very small value.

3.2 Training Loop

In this paper, 80% of the data is randomly selected as the training set, while the remaining 20% is used as the test set to validate the model. The network parameters are detailed in Table 1. The optimizers in GP and CNN are both based on gradient descent for weight updates. Here, the widely used BFGS algorithm is used to implement this optimization process. The Matern kernel function is chosen in GP algorithms, which is a generalization of the radial basis function (RBF) kernel. Target-value normalization refers to standardizing the target value of y by removing the mean and scaling to the unit variance, which is recommended when using a zero-mean, unit-variance prior.

Fig. 3 shows the loss of the validation set for MLP, CNN, GP, and CNN-GP algorithms. The inversion errors of MLP, CNN, and CNN-GP algorithms decrease as the number of training times increases at a signal-to-noise ratio (SNR) of 25dB. CNN-GP algorithm converges faster than the CNN and MLP algorithms, with smoother fluctuations in loss. The overall loss error of the CNN-GP algorithm is smaller than that of the CNN algorithm in all three directions with a higher inversion accuracy. The CNN-GP algorithm converges gradually around the 230th iteration. However, after approximately 370 iterations, the inversion error increases due to model overfitting. MLP and CNN algorithms achieve the lowest inversion error at around 270 iterations. The GP algorithm shows relatively larger inversion errors, with an MAE of 0.067, 0.059, and 0.135 for inversion in the x , y , and z directions, respectively. This indicates lower inversion accuracy compared to the converged MLP, CNN, and CNN-GP algorithms.

Table 1. Parameter settings of the inversion algorithms

Hyperparameter	CNN	GP	CNN-GP	MLP
Training set	960	960	960	960
Test set	240	240	240	240
activation func-	LeakyReLU	-	LeakyReLU	LeakyReLU
optimizer	Adam	BFGS	Adam, BFGS	Adam
kernel function	-	Matern	Matern	-
Normalized	-	False	False	-
SNR (dB)	25	25	25	25

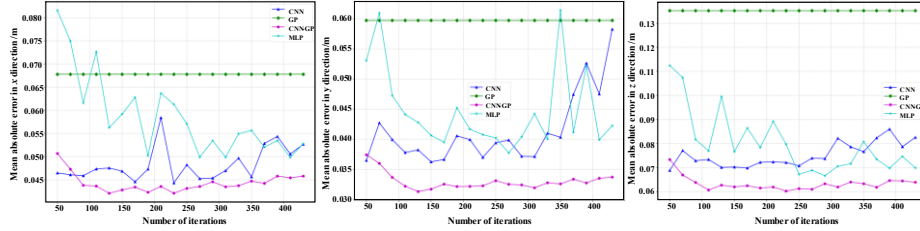


Fig. 3. Changes in training loss of the CNN, GP, MLP, and CNN-GP algorithms.

Table 2. Grid search algorithm and its search domain.

Hyperparameter	Candidate values for hyperparameters			
	CNN	GP	CNN-GP	MLP
Number of iterations	[350,400,450,500,550]	-	[350,400, 450,500,550]	
Activation function	[Leaky ReLU, ELU, ReLU, Sigmoid, Tanh]	-	[Leaky ReLU, ELU, ReLU, Sigmoid, Tanh]	
Optimizer	Adam	BFGS	Adam, BFGS	Adam
Kernel function	-	[Matern, WhiteKernel, RBF, ConstantKernel, Matern + WhiteKernel, ConstantKernel +RBF]		-
Number of optimizer restarts	-	[0, 5, 10, 15]		-
Normalized	-	[False, True]		-
No. of convolution layers	[1, 2, 3]	-	[1, 2, 3]	-
Pooling layer	[False, True]	-	[False,True]	-
Number of FC layers	[1, 2, 3, 4]	-	[1, 2, 3, 4]	
Sampling space	[0.1, 0.2, 0.3, 0.4, 0.5]			

3.3 Model Optimization

Grid Search. Algorithm Grid search algorithm is used to determine the optimal combination of hyperparameters by exhaustively searching through the user-defined hyperparameter space. It is suitable for scenarios where multiple hyperparameters are limited in the space to search. Parallelization of the grid search algorithm can increase the efficiency, allowing faster identification of the optimal hyperparameter combination.

However, using the grid search algorithm in high-dimensional spaces may lead to the problem of dimensionality catastrophe. This is because the search space will grow exponentially with increasing dimensionality, which makes it necessary to limit the sampling range. Therefore, to avoid an excessive number of configurations and ensure efficient utilization of computational resources, it is feasible to define a smaller search space and adjust no more than three hyperparameters. The grid search algorithm is usually implemented with nested loops, iterating through all possible combinations of

Table 3. Bayesian search algorithm and its search domain.

Hyperparameter	Candidate values for hyperparameters			
	CNN	GP	CNN-GP	MLP
Number of iterations	(350, 550)	-	(350, 550)	(350, 550)
Activation function	(0, 4.9)	-	(0, 4.9)	(0, 4.9)
Optimizer	Adam	BFGS	Adam, BFGS	Adam
Kernel function	-	(0, 6.9)	(0, 6.9)	-
Normalized	-	(0, 1.9)	(0, 1.9)	-
No. of convolutional layers	(1, 3.9)	-	(1, 3.9)	-
Pooling layer	(0, 1.9)	-	(0, 1.9)	-
Number of FC layers	(1, 4.9)	-	(1, 4.9)	(1, 4.9)
Sampling space	(0.1, 0.5)	(0.1, 0.5)	(0.1, 0.5)	(0.1, 0.5)

candidate values for each hyperparameter to train the models and select the best one. Table 2 shows the hyperparameters that need to be adjusted when using the grid search algorithm.

Bayesian Search Algorithm. Compared to the grid search algorithm, the Bayesian search algorithm offers higher computational efficiency and requires fewer trials. Its probabilistic surrogate model is built using evaluations from previous experiments. To avoid getting trapped in local optima, the model is updated at each trial, and the next sampling point is chosen to balance exploration and exploitation. BO algorithm consists of two key components: the Bayesian surrogate model for modeling the objective function and the acquisition function for determining the next sampling point.

The Bayesian search algorithm maintains consistency with the grid search algorithm in terms of hyperparameter selection but differs in candidate values. The Bayesian search algorithm uses continuous values, which must be specified within a range. The specific parameters are shown in Table 3. Of the hyperparameters listed above, only Sampling space is a real number, while the others are integers. Therefore, before conducting model training, these parameters must be rounded down. For Activation Function and Kernel Function, the integer values must correspond to the relevant candidate values in Table 2. For example, if the hyperparameter of Kernel Function in Table 3 rounds down to 0, then its corresponding value during training corresponds to the Matern function.

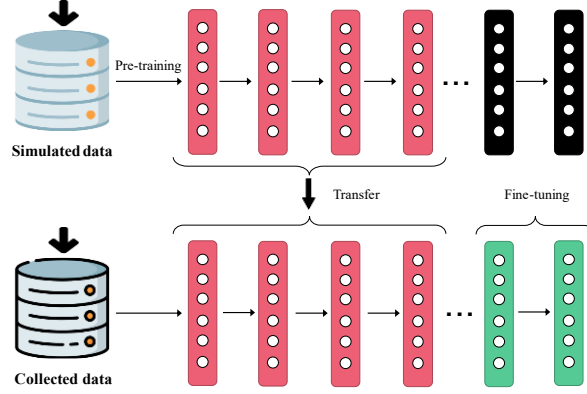


Fig. 4. Model-based transfer learning for underground metal targets.

3.4 Transfer Learning

This paper conducts model training based on simulated data, resulting in satisfactory outcomes. To efficiently obtain a model with good inversion performance for real collected data based on the existing model, it is necessary to perform model transfer. This paper uses model-based transfer learning for two main reasons. Firstly, the response data generated by both simulation and real detection are generally similar due to their shared detection scenario. Therefore, the feature extraction part in CNN-GP can be smoothly transferred, and only a slight training on the part after the flattening layer with real response data is needed to obtain an inversion model applicable to real scenarios. Secondly, data collection in real scenarios is slow, making it difficult to generate sufficient data for model training. Hence, pre-training with simulated data is necessary, followed by fine-tuning with real data.

Fig. 4 displays the convolutional and pooling layers in red, which have been pre-trained using a large amount of simulated data. Once training is completed, this part is transferred to the network training for the collected data. The feature extraction layers for the collected data are then initialized with effective weights and biases obtained from the pre-training process. Only a few training iterations are required to fit the collected data, achieving model transfer. The green part in Fig. 4 is the MLP, which is initialized normally and updated with respective weights and biases during training using the collected data.

4 Result and Discussion

4.1 Dataset

In this paper, the length of the metal target is 0.4m, the radius is 0.2m, and the pitch and roll angles are 0 and 90 degrees, respectively. The magnetic field intensity in the three axial directions of the metal target is shown in Fig. 5. Electromagnetic signals are emitted from multiple locations, and the intensity information of the secondary field signals collected at these locations is used as observation data. From the heatmap of the magnetic field intensity in the z -axis direction, it can be observed that the value of the magnetic field induction observation data is greater at positions above the target.

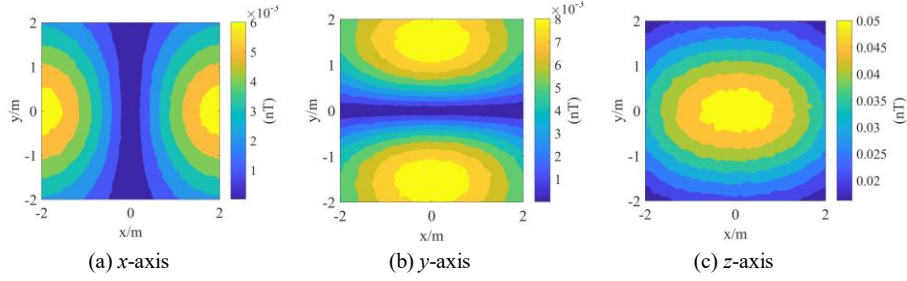


Fig. 5. Heatmaps of magnetic field intensity in the three axial directions of the metal target

The inversion algorithm in this paper is entirely implemented in the Python programming language, and Miniconda is used for managing and deploying the runtime environment.

4.2 Model Inversion and Result Analysis

Fig. 6 lists the results of the four algorithms with different SNRs. The inversion errors increase gradually as the SNRs decrease, and the most affected one is the inversion effect of the target in the z direction. At a SNR of 10dB, the inversion errors of MLP, CNN, and CNN-GP algorithms approach approximately 0.2m, while the errors in the x and y directions are around 0.06m. Regarding the GP algorithm, its overall inversion errors are relatively large. Under a SNR of 10dB, its error in the z direction reaches 0.6m, with similarly 0.1m errors in the x and y directions. This is significantly inferior to the models trained by the CNN and CNN-GP algorithms. Overall, the inversion performance of the CNN-GP algorithm appears superior to that of the MLP and CNN algorithms.

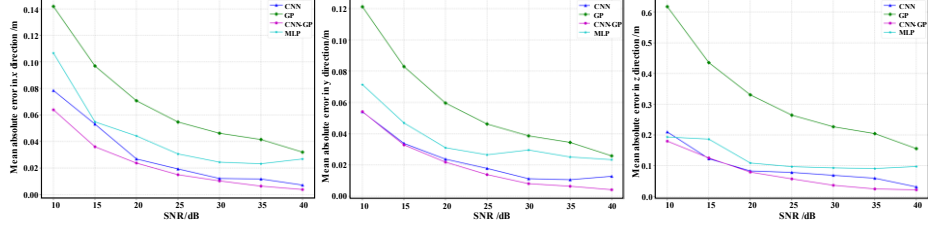


Fig. 6. Inversion errors of four algorithms under different SNRs.

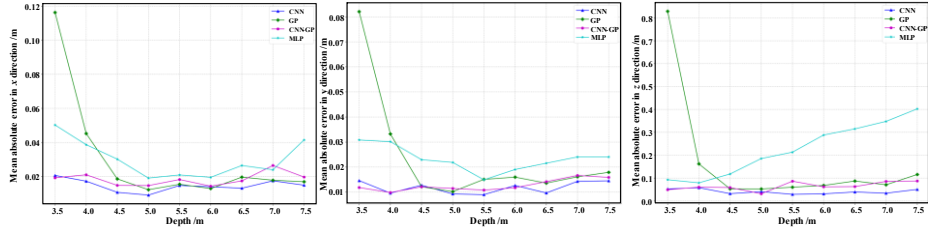


Fig. 7. Inversion errors of four algorithms at different depths.

Fig. 7 shows the inversion errors of the four models for cylindrical metal objects at different buried depths. It can be observed that the GP algorithm exhibits poor inversion accuracy for targets at depths ranging from 3.5m to 4.5m. For targets at depths ranging from 4.5m to 7.5m, the four algorithms maintain inversion errors in the x and y directions at around 2cm. However, depth significantly affects prediction accuracy in the z direction, particularly for the CNN-GP and GP algorithms. Notably, targets at depths ranging from 6.5m to 7.5m exhibit a marked increase in inversion errors. The CNN algorithm is less affected by depth; even with increasing buried depth, its inversion error growth remains relatively gradual, resulting in better overall performance compared to the CNN-GP and GP algorithms.

Table 4. Performance comparison of two search algorithms

Algorithms	Grid Search		Bayesian Search	
	Iterations	Time/h	Iterations	Time/h
CNN	3000	797	120	10
GP	240	\ll 797	120	3
CNN-GP	720000	\ll 797	120	13
MLP	600	\ll 797	120	4.5

4.3 Comparative Analysis of Model Optimization

Table 4 shows that the time consumption for the grid search algorithm is significantly higher than the Bayesian search algorithm. Therefore, the Bayesian search algorithm proves to be more convenient and faster for conducting experiments. The average

inversion errors of the algorithms before and after optimization using the Bayesian search algorithm are presented in Table 5. It can be observed that after Bayesian search algorithms, the inversion performance of the four algorithms improves for different SNRs in terms of the x , y , and z directions. Specifically, the CNN-GP algorithm exhibits higher inversion accuracy compared to other algorithms in scenarios with higher SNRs. However, under low SNRs, its predictions for the z direction are less accurate compared to MLP and CNN algorithms. Nevertheless, its inversion performance for the x and y directions surpasses that of other algorithms.

Table 5. Error comparison of the inversion algorithms under different SNRs before and after optimization

Algorithms	SNR/dB	10	15	20	25	30	35	Noise-less	
CNN	-	X/m	0.0784	0.0531	0.0268	0.0193	0.0121	0.0117	0.0071
		Y/m	0.0539	0.0337	0.0237	0.0177	0.0111	0.0105	0.0127
		Z/m	0.2098	0.1227	0.0829	0.078	0.0686	0.0591	0.031
	Optimization	X/m	0.0603	0.0315	0.0182	0.0113	0.0074	0.0067	0.0069
		Y/m	0.047	0.0278	0.0203	0.0094	0.0071	0.005	0.0037
		Z/m	0.1236	0.0728	0.0536	0.0369	0.0305	0.0195	0.0133
GP	-	X/m	0.1418	0.0969	0.0707	0.0546	0.0462	0.0414	0.0319
		Y/m	0.1212	0.0829	0.0595	0.0462	0.0385	0.0344	0.0258
		Z/m	0.6176	0.4355	0.3306	0.2644	0.2267	0.2044	0.1554
	Optimization	X/m	0.0913	0.0615	0.0456	0.0374	0.0336	0.0322	0.0292
		Y/m	0.0776	0.0528	0.0396	0.0317	0.0281	0.0265	0.0238
		Z/m	0.2171	0.2314	0.1942	0.1671	0.1534	0.1492	0.1394
MLP	-	X/m	0.1066	0.0548	0.0442	0.0306	0.0244	0.0232	0.0268
		Y/m	0.0713	0.0469	0.031	0.0265	0.0295	0.0251	0.0233
		Z/m	0.1933	0.1864	0.1088	0.0969	0.0929	0.0907	0.0974
	Optimization	X/m	0.0537	0.0316	0.0187	0.0126	0.0151	0.009	0.0049
		Y/m	0.046	0.0385	0.0217	0.0101	0.0092	0.0082	0.0043
		Z/m	0.0743	0.0622	0.0482	0.0358	0.0238	0.0201	0.0127
CNN-GP	-	X/m	0.0639	0.036	0.0237	0.0149	0.0102	0.0063	0.0038
		Y/m	0.0541	0.0328	0.0218	0.0138	0.008	0.0064	0.0041
		Z/m	0.1798	0.1252	0.079	0.0569	0.0362	0.0243	0.0215
	Optimization	X/m	0.0488	0.0285	0.0191	0.0116	0.0068	0.0047	0.0004
		Y/m	0.0412	0.0249	0.0147	0.0102	0.007	0.0033	0.0005
		Z/m	0.1528	0.093	0.0513	0.0372	0.025	0.0137	0.0011

Table 6. Comparison of the inversion average errors of the four algorithms at different depths before and after optimization

Depth/m	-			Optimization		
	X/m	Y/m	Z/m	X/m	Y/m	Z/m
CNN	0.0196	0.0165	0.0517	0.0134	0.0143	0.0518
GP	0.0886	0.0726	0.5714	0.0680	0.0582	0.4184
CNN-GP	0.0192	0.0140	0.0740	0.0120	0.0098	0.0448
MLP	0.0360	0.0291	0.2189	0.0122	0.0120	0.0414

The results of the average error of four algorithms for cylindrical metal objects buried at different depths are shown in Table 6. It can be observed that except for the CNN algorithm in the z direction, the other algorithms demonstrate improved inversion performance for the x , y , and z directions post-optimization. It is evident that the Bayesian search algorithm effectively reduces the overall average inversion error for targets buried at different depths.

5 Discussion

This paper aims to obtain the three-dimensional position of underground metal targets from electromagnetic data using the CNN-GP algorithm. Typically, electromagnetic data can be acquired through either the frequency domain electromagnetic (FDEM) method or the time domain electromagnetic (TDEM) method. The FDEM method emits low-frequency continuous electromagnetic waves and analyzes the amplitude and phase of the secondary field signals received at multiple locations to obtain the position of the target. The TDEM method determines the parameters of the target by emitting pulse currents and analyzing the secondary field decay signals. Compared to the TDEM method, the FDEM method does not need to account for the decay characteristics of secondary signals over time, thereby reducing the impact of soil on detection. Hence, this paper generates electromagnetic data based on the FDEM method and forward modeling. Experimental results demonstrate that the CNN-GP algorithm can quickly obtain inversion results within 0.2s, meeting the real-time and rapid requirements of underground metal target detection.

Although FDEM data can provide more direct information on resistivity and conductivity, the lack of temporal characteristics in electromagnetic data limits the application of methods that require temporal or spatial correlations, such as RNN, LSTM, and Transformer models. In this paper, preliminary experiments indicate that these methods struggle to make accurate predictions. Therefore, in complex applications requiring information on the material, shape, or orientation of underground metal targets, the time domain electromagnetic (TDEM) method becomes the preferred choice, offering possibilities for the application of RNN methods. Additionally, when extracting features from the secondary field magnetic map, the magnetic field intensity at each location is only related to its position and is not influenced by surrounding positions. The Transformer method requires positional context in the input data and is commonly

used in natural language processing. Thus, future research will focus on further investigating the temporal and spatial characteristics of FDEM data, and analyzing the feasibility of LSTM, Transformer models, and other methods in underground metal target detection applications. Future work will continue to explore deeper research on LSTM and Transformer methods based on CNN to improve the estimation accuracy of parameters such as the material properties of metal targets.

In addition, we have analyzed the performance of four algorithms under different amounts of data, and the results show that as the amount of data increases, the errors decrease in the x and y directions. However, there is no significant reduction in the z direction due to the increase in the acquisition space. Therefore, it can be concluded that the inversion effects of the four algorithms on the x and y directions are sensitive to the amount of data, while for the z direction, only the GP algorithm and the amount of data show a positive correlation. MLP, CNN, and CNN-GP algorithms cannot improve the inversion effect in the z direction by increasing the amount of data.

6 Conclusion

This paper proposes the CNN-GP algorithm for underground metal target detection based on the forward model and convolutional inversion theory. The proposed algorithm is compared with the MLP, CNN, and GP algorithms, and all four algorithms demonstrate high inversion efficiency suitable for real-time detection and inversion, with inversion results obtained within 1s. The CNN-GP algorithm outperforms the other algorithms in terms of the average inversion error. The impact of depth on the GP algorithm results in significant error fluctuations. In contrast, the CNN and CNN-GP algorithms are less sensitive to depth when estimating in the x and y directions. The CNN algorithm performs relatively better when inverting deeper objects, demonstrating stable predictions for targets at different depths with minimal sensitivity to depth. To improve the performance of inversion algorithms, this paper uses Bayesian search algorithms to find the optimal network structure. Furthermore, this paper addresses the discrepancy between simulated and real data through transfer learning. The feature extraction part related to CNN can be retained, and only the part of the model that needs to be regressed can be retrained using real data. This training process does not require multiple iterations or a large amount of data, which reduces the time of real data model training and improves model performance.

Acknowledgments. This work was funded by National Natural Science Foundation of China (no. 61971031).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Liu, C., Liu, S., Li, Z., Wang, L., Kan, S., Liang, J.: Research on ground-airborne frequency-domain electromagnetic rapid imaging method based on space magnetic gradient anomaly. *IEEE Transactions on Geoscience and Remote Sensing* (2023)
2. Banning, E.B.: The archaeological impacts of metal detecting. *Open Archaeology* 5(1), 180–186 (2019)
3. Lei, W., Luo, J., Hou, F., Xu, L., Wang, R., Jiang, X.: Underground cylindrical objects detection and diameter identification in gpr b-scans via the cnn-lstm framework. *Electronics* 9(11), 1804 (2020)
4. Zhang, Y., Bian, L., Xu, Z., He, Y., Qin, T., Huang, Z.: Dual-mode detection method for uxo targets by measuring magnetic anomaly and electromagnetic response with fenimosib/pzt-5a composite. *IEEE Transactions on Geoscience and Remote Sensing* (2024)
5. Jianguo, S., Ntibahanana, M.: Developing deep learning methods for pre-stack seismic data inversion. *Journal of Applied Geophysics* 222, 105336 (2024)
6. Kang, M.S., Kim, N., Lee, J.J., An, Y.K.: Deep learning-based automated underground cavity detection using three-dimensional ground penetrating radar. *Structural Health Monitoring* 19(1), 173–185 (2020)
7. Wan, Y., Wang, Z., Wang, P., Zhang, C., Duan, S., Li, N.: A comparative study of inversion optimization algorithms for underground metal target detection. *IEEE Access* 8, 126401–126413 (2020)
8. Chávez, C.E., Alonzo-Atienza, F., Alvarez, D.: Avoiding the inverse crime in the inverse problem of electrocardiography: estimating the shape and location of cardiac ischemia. In: *Computing in Cardiology 2013*. pp. 687–690. IEEE (2013)
9. Puzyrev, V., Swidinsky, A.: Inversion of 1d frequency-and time-domain electromagnetic data with convolutional neural networks. *Computers & geosciences* 149, 104681 (2021)
10. Moghadas, D.: One-dimensional deep learning inversion of electromagnetic induction data using convolutional neural network. *Geophysical Journal International* 222(1), 247–259 (2020)
11. Puzyrev, V.: Deep learning electromagnetic inversion with convolutional neural networks. *Geophysical Journal International* 218(2), 817–832 (2019)
12. Aleardi, M., Vinciguerra, A., Hojat, A.: A convolutional neural network approach to electrical resistivity tomography. *Journal of Applied Geophysics* 193, 104434 (2021)
13. McNeeill, J.: Electromagnetic terrain conductivity measurement at low induction numbers. *Geonics technical note TN-6* (1980)
14. Huang, H., SanFilipo, B., Won, I.: Planetary exploration using a small electromagnetic sensor. *IEEE transactions on geoscience and remote sensing* 43(7), 1499–1506 (2005)
15. Manstein, Y., Manstein, A.: Emi sensor nemfis: method, equipment and case stories of archaeological prospection. *ArcheoSciences. Revue d'archéométrie* (33 (suppl.)), 321–324 (2009)
16. Nabighian, M.N.: *Electromagnetic methods in applied geophysics: Voume 1, theory*. Society of Exploration Geophysicists (1988)
17. Qu, X., Li, Y., Fang, G., Yin, H.: A portable frequency domain electromagnetic system for shallow metal targets detection. *Progress In Electromagnetics Research M* 53, 167–175 (2017)
18. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12(7) (2011)
19. Tieleman, T.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2), 26 (2012)

20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: *Neural networks: Tricks of the trade: Second edition*, pp. 437–478. Springer (2012)
22. Smith, L.N.: Cyclical learning rates for training neural networks. In: *2017 IEEE winter conference on applications of computer vision (WACV)*. pp. 464–472. IEEE (2017)
23. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. *Neural computation* 18(7), 1527–1554 (2006)
24. George, D., Shen, H., Huerta, E.: Deep transfer learning: A new deep learning glitch classification method for advanced ligo. arXiv preprint arXiv:1706.07446 (2017)
25. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1717–1724 (2014)
26. Huang, J.T., Li, J., Yu, D., Deng, L., Gong, Y.: Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In: *2013 IEEE international conference on acoustics, speech and signal processing*. pp. 7304–7308. IEEE (2013)